commodore
COMPUTER CLUB
22
L. 3.000
La rivista degli utenti di sistemi Commodore
Mensile - 25 luglio/25 agosto 1985 - Anno IV - n. 22 - Sped. Abb. Post. Gr. III/70 CR - Distr. MePe
SPECIALE VACANZE
Tutti i giochi
da fare sulla spiaggia
Simula un computer
con conchiglie e sassolini
La scatola robot
che impara a giocare
I quiz sotto l'ombrellone
Giochi, giochi, giochi
per Vic 20, C 16, C 64
systems

Un'iniziativa condotta con la nota rivista Computer

PROGRAMMO IN BASIC

Il linguaggio del futuro in un manuale rapido e completo di Clizio Merli
pagg. 224 (L. 9.000)

Il Basic, attualmente il linguaggio più conosciuto - adatto all'utilizzo su qualunque tipo di macchina e in particolare sul personal e gli home-computer - può essere appreso in poche ore con l'ausilio di questo agile manuale.

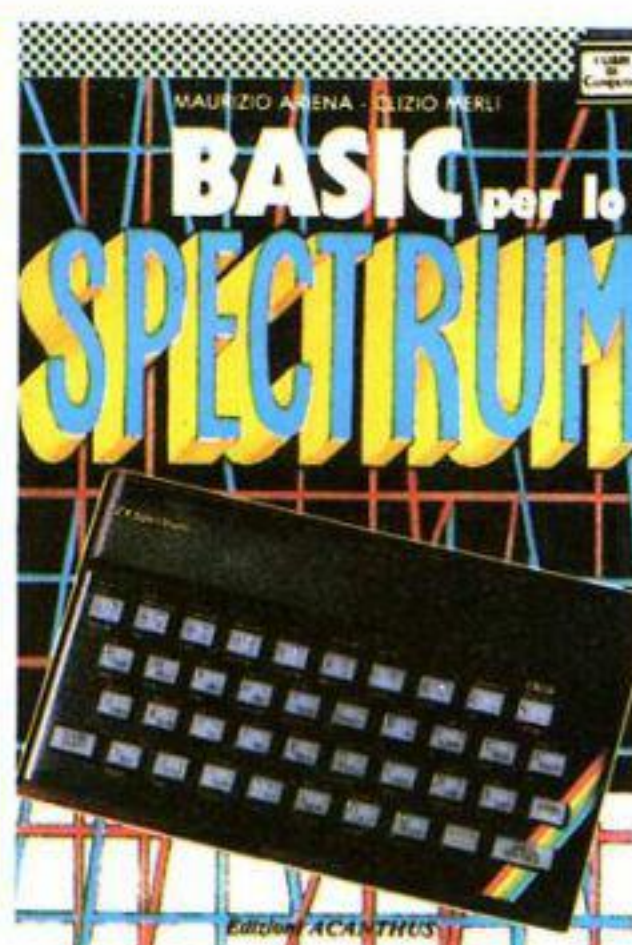
COME SCEGLIERE UN COMPUTER

Guida pratica per l'acquisto di un mini o di un micro computer professionale di Michele Di Pisa
pagg. 160 (L. 6.000)

Quale modello scegliere tra gli oltre 600 computer commercializzati in Italia? La conoscenza delle caratteristiche delle varie macchine è indispensabile. Con un approccio a "menu" l'Autore vuol essere guida proprio in questa fase.

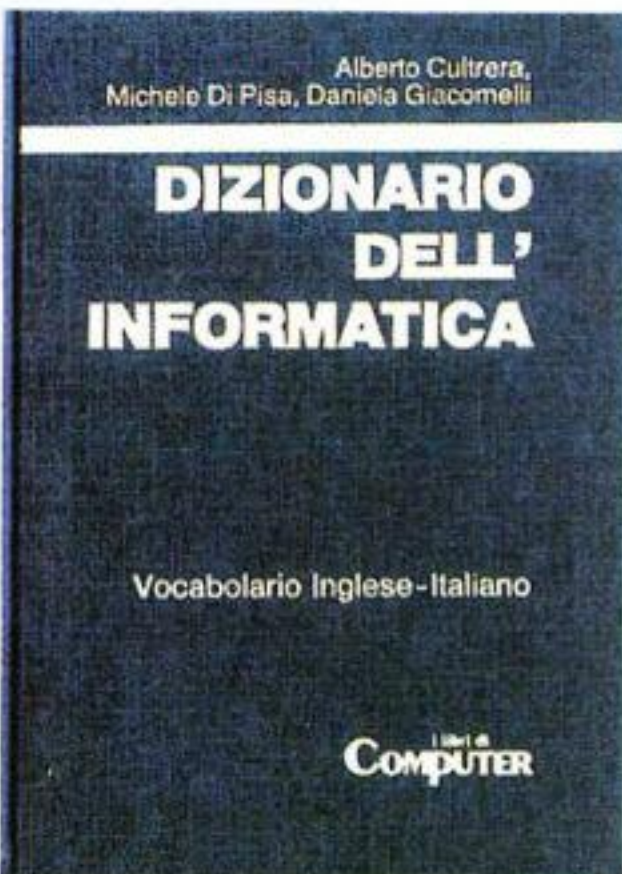
UTILITY E ROUTINE PER IL COMMODORE 64

di Gloriano Rossi
pagg. 192 (L. 9.000)

L'esecuzione di una istruzione BASIC può richiedere diverse centinaia di passi di programmi in linguaggio macchina. La dimensione dei programmi è ciò che intimidisce maggiormente l'utilizzatore medio di Commodore: aiutato da questo testo chiunque potrà affrontare senza problemi il processo di scrittura di un programma.

BASIC PER LO SPECTRUM

di Maurizio Ariena e Clizio Merli
pagg. 192 (L. 9.000)

Un libro per quanti hanno acquistato il computer ZX Spectrum della Sinclair e intendono sfruttarne appieno tutte le capacità, dall'hardware alla programmazione in assembly (linguaggio macchina).

I volumi, che sono comunque in vendita nelle migliori librerie di tutta Italia, possono anche essere richiesti direttamente all'Editore.
Importante: l'ordine minimo dovrà essere di L. 15.000.

DIZIONARIO DELL'INFORMATICA

Vocabolario Inglese-Italiano di Cultrera, Di Pisa, Giacomelli
pagg. 388 (L. 25.000)

Uno strumento indispensabile per chi si avvicina al mondo dell'informatica e per gli specialisti che hanno l'esigenza di accedere alla dinamica letteratura anglosassone.

Edizioni ACANTHUS

VIALE GRAN SASSO, 23 - 20131 MILANO

Inviatemi i seguenti volumi:

Titolo	quantità	prezzo unitario
spese postali		L. 2.000
	totale	L.

Pagherò contrassegno il dovuto (più L. 2.000 per contributo spese postali) al ricevimento. Potrò restituire i libri entro 8 giorni se non saranno di mio gradimento e avere il rimborso immediato.

COGNOME

NOME

VIA N.

C.A.P. CITTÀ

FIRMA

DATA

Scrivere in stampatello e spedire in busta chiusa.

Sommario

SPECIALE VACANZE

Direttore: Alessandro de Simone
Redazione/collaboratori: Giovanni Bellù, Andrea e Alberto Boriani, Giancarlo Castagna, Eugenio Coppari, Marco De Martino, Luca Galluzzi, Giancarlo Mariani, Flavio Molinari, Enrico Scelsa, D. Matturro, M.L. Nitti, Massimo Patti, Carla Rampi, Fabio Sorgato, Danilo Toma
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Ufficio Grafico: Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Mirco Croce (coordinatore), Giuseppe Porzani, Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio - Segretaria: Lilliana Degiorgi
● Roma: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Abbonamenti: Marina Vantini
Tariffe: prezzo per copia L. 3.000. Abbonamento annuo (11 fascicoli) L. 28.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 55.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl **Fotolito:** Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/1/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distribuzione: MePe, via G. Carcano 32 - Milano

Commodore 64, Commodore 128 & CP/M

☐ **Che cosa è il sistema operativo del Commodore 64 e, in particolare, che cosa è il sistema operativo CP/M e come lo si può utilizzare, inserendo l'apposita scheda, col Commodore 64?** *(Francesco Bellini - Firenze)*

• La tua domanda capita a fagiolo, data la presentazione dei nuovi modelli Commodore, e ti ringraziamo per averci dato lo spunto per parlarne. Cercheremo di essere brevi, ma chiari. Per Sistema Operativo (S.O.) si intende quel particolare complesso di istruzioni e comandi in linguaggio macchina (L.M.) che consentono al calcolatore di eseguire programmi in vari linguaggi (come il Basic), routine di memorizzazione, output su stampante, collegamenti con modem, periferiche varie eccetera.

Ogni computer ha il proprio sistema operativo che, molto spesso, è incompatibile con altre marche o, addirittura, con modelli della stessa casa.

Ciò vuol dire che quel particolare gruppo di istruzioni ha un significato ben preciso solo per quel modello e non per altri.

Quasi sempre, però, il S.O. è, come si dice in gergo, "trasparente", vale a dire che l'utente finale può non accorgersi minimamente che è in funzione un particolare sistema operativo e molti lettori, come te, non sanno neppure che esiste!

In altre parole, se ti trovi davanti un computer che "parla" Basic (e non sai altro di lui) l'istruzione PRINT"PIPPO" produrrà lo stesso effetto sia che il computer sia un Vic 20, oppure uno Spectrum oppure un Olivetti o un qualsiasi altro computer che parli il Basic.

Ciò significa che il linguaggio Basic è "guidato" dal sistema operativo del calcolatore che, nel complesso, provvede a produrre lo stesso effetto nei vari computer.

Chiarito che una cosa è il linguaggio e ben altra è il S.O., c'è da dire che il sistema operativo cambia profondamente a seconda del microprocessore (= cervello) adoperato sul computer in oggetto.

I computer Commodore utilizzano i micro 6502, da cui deriva il 6510 del C-64, che gestiscono i dati secondo una particolare "architettura".

Un altro grande microprocessore è il famoso Z-80, col quale è possibile strutturare un'architettura diversa.

Quando si realizzò lo Z-80, una equipe di tecnici, sfruttando e ottimizzando le interessanti caratteristiche di tale micro, idearono un sistema ottimizzato di organizzazione di dati e memoria che prese il nome di CP/M (Control Programs for Microcomputers). Molte case costruttrici di computer si gettarono a capofitto col CP/M ed il risultato è che oggigiorno è disponibile una biblioteca di programmi incredibilmente vasta grazie alla compatibilità di macchine che, pur diverse tra loro, possono comunicare proprio in quanto posseggono lo stesso sistema operativo.

Qualche tempo dopo si realizzò il micro 6502, che non ebbe lo stesso successo dello Z-80, ma consentì di produrre piccoli computer a basso costo. Tra questi figurarono i Commodore, gli Apple, gli Atari e tanti altri. Purtroppo ognuno di questi, pur possedendo l'identico micro, avevano diversi sistemi operativi e l'incompatibilità presentò subito il vantaggio della separazione delle sfere commerciali, ma bloccò, di fatto, il "decollo" del 6502 in modo da opporre una seria barriera concorrenziale al CP/M.

In questi ultimi tempi un nuovo microprocessore, il 68000 della Motorola, sta prendendo piede soprattutto dopo le positive esperienze con l'8088, che deve essere considerato il "vero" precursore dell'MS/DOS, e renderà tra breve (pettegolezzi di portineria) obsoleto qualsiasi altro sistema, CP/M compreso. Il 68000, in effetti, consente un'architettura potente, versatile e veloce che è sfruttata dal nuovo astro: l'MS/DOS.

Riassumendo: oggi, 1985, sono disponibili due veri grandi sistemi operativi:

CP/M, vecchio ma che vanta una straordinaria biblioteca di programmi soprattutto per le aziende;

MS/DOS, il sistema operativo del futuro che però, data la giovane età, non dispone di programmi che sfruttino in pieno le notevoli potenzialità del nuovo 68000.

Oltre ai due prima citati vi sono i vari sistemi Commodore, Apple, eccetera, incompatibili tra loro ma pur dotati di una grande biblioteca soprattutto destinata alla casa, ai giovani, alle piccole e piccolissime aziende.

Poichè non è facilmente possibile "simulare" col 6502 il micro Z-80, ecco che si rende necessaria una scheda elettronica aggiuntiva che, di fatto, esclude il 6502 sostituendolo con lo Z-80, presente sulla stessa scheda. In pratica ciò che rimane del 64 è la tastiera e la memoria RAM che viene ora gestita dal nuovo microprocessore ospite.

La scheda CP/M da collegare col C-64 non sembra aver goduto molto successo a causa di alcune presunte incompatibilità inevitabili per mancanza di accordi sullo sfruttamento del brevetto. Con l'esplosione del fenomeno MS/DOS e con l'esigenza di restare sulla cresta dell'onda, la Commodore ha avuto una bella "pensata" ed ha telefonato alla Digital (che detiene il brevetto CP/M) dicendo più o meno così:

"Il CP/M tra un pò cola a picco e le pretese di royalty del vostro brevetto devono essere ridimensionate. Se voi ci date il CP/M a basso costo, noi, col nome Commodore, contribuiamo a tenere in vita il vostro sistema e avete il tempo per realizzare qualcosa d'altro. Noi, viceversa, non siamo costretti ad abbandonare l'enorme esperienza accumulata con microprocessori a soli otto bit e presentiamo un apparecchio a basso costo abbastanza innovativo".

Probabilmente, dall'altro capo del filo, hanno risposto, forse a denti stretti, "occhei" ed il risultato è un apparecchio (il Commodore 128) che, in effetti, è una fusione di tre computer su di una stessa piastra:

a/ Commodore 64, identico in tutto e per tutto al C-64 che si trova in commercio.

b/ Commodore 128, il nuovo apparecchio che dispone di potenti comandi Basic e di 128K di RAM (un C64 PLUS-PLUS!)

c/ CP/M che consente di accedere all'immensa biblioteca disponibile sul nostro pianeta.

Ogni sezione elettronica ha il proprio micro: 6502 per la sezione C-64, 8502 per il C-128 e Z-80 per il CP/M.

Oltre a questi cervelli sono presenti altri chip particolari che consentono, soprattutto, la selezione della parte elettronica che interessa.

Il risultato di tutto questo è che, grazie al C-128, che ci auguriamo di vedere al prossimo SMAU anche in Italia, la Commodore renderà disponibile, con il classico uovo di Colombo, un modello che elimina con un sol colpo il problema della incompatibilità col modello precedente (C-64) e col S.O. attualmente più utilizzato (CP/M).

Mettendo, nello stesso contenitore ben tre computer al prezzo di uno solo di essi, il successo di vendite non potrà che essere brillante.

Incompatibilità col Turbo-tape

☐ **Usando il programma Turbo-tape, allocato, come è noto, nella parte "alta" della memoria, si crea incompatibilità con altri programmi, specie in Linguaggio Macchina (es. routine grafiche di Toma), poichè occupano le stesse locazioni di memoria. C'è un modo per "spostare" il Turbo-tape in altra zona di memoria?** *(Enzo Filippi - Albiano-TN)*

• Spostare un programma in Linguaggio Macchina, quale il Turbo-tape, è possibile, ma richiede un notevole studio degli indirizzamenti assoluti ed un'attenta analisi della memorizzazione di variabili di comodo.

Spieghiamoci meglio.

Supponiamo che il programma in L.M. sia memorizzato dalla locazione 50000 alla 52000 e che lo riallochiamo, con una semplice routine, dalla locazione 40000 alla 42000.

Supponiamo anche che consenta al suo interno un'istruzione del tipo:
STA - 51900

Questa istruzione, come è noto, deposita il contenuto dell'accumulatore nella locazione decimale 51900. Un puro e semplice "trasferimento" del programma a partire, come già detto, dalla locazione 40000, non modifica l'indirizzo assoluto dell'istruzione con la conseguenza che il programma, benchè "riallocato" in altra zona di memoria, continuerà a trascrivere il dato contenuto nell'accumulatore nella locazione 51900, alterando una zona di memoria a torto ritenuta libera dopo l'operazione di trasferimento.

In altre parole, chi desidera effettuare riallocazioni deve disassemblare la routine in L.M. ed esaminare (ovviamente modificandole) tutte le istruzioni con indirizzo assoluto, quelle indicizzate, rintracciare eventuali tabelle di messaggi e... sperare che le modifiche apportate siano quelle giuste e idonee!

Vuoi le istruzioni?

□ **Come posso fare affinchè, prima dell'inizio di un gioco, vengano scritte le regole dello stesso?** *(Dario Gabrielli - Cadoneghe - PD)*

• Supponiamo, per semplificare le cose, che il gioco vero e proprio inizi alla riga Basic 200 e termini alla 10000.

Aggiungi le seguenti linee:

```
100 PRINT"VUOI LE ISTRUZIONI? (S/N)"
110 GET A$:IF A$T"" THEN 110
120 IF A$="S" THEN GOSUB 10100
200 REM QUI INIZIA IL GIOCO
.............
.............
10000 REM QUI TERMINA IL TUO GIOCO
10100 PRINT"IL GIOCO CONSISTE NEL"
10110 PRINT"DISPORRE LE CARTE DA"
10120 PRINT"GIOCO IN MODO CHE"
.....(eccetera).....
11000 RETURN
```

La riga 100 scrive un messaggio al quale si deve rispondere premendo il tasto "S" oppure "N".

La 110 attende che venga premuto un tasto.

La riga 120 esamina se il tasto premuto è una "S". In questo caso va ad eseguire il sottoprogramma (GOSUB) che inizia alla riga 10100 (N.B. dopo la fine del gioco vero e proprio).

Dalla riga 200 alla 10000, come abbiamo ipotizzato, è presente il tuo gioco che non è necessario modificare in alcun modo.

Alla riga 10100 inizia il sottoprogramma, richiamato dalla riga 120 se hai premuto "S", che non è altro che un gruppo di istruzioni di PRINT che hanno il compito di visualizzare le regole del gioco.

L'ultima riga del sottoprogramma deve essere costituita dall'istruzione RETURN che comunica al calcolatore l'ordine di ritornare... da dove era venuto (riga 120) e di riprendere il programma dall'istruzione sucessiva al GOSUB.

Ma l'istruzione successiva è costituita proprio dall'inizio del gioco vero e proprio che, appunto, parte immediatamente al termine della visualizzazione delle regole del gioco.

Strani codici

□ **Ho scoperto casualmente un nuovo simbolo che si riscontra sia sul C-64 che sul Vic 20 e C-16. Esso si ottiene, dopo aver aperto le virgolette, premendo i tasti CTRL e quello di freccia a sinistra. Il risultato è una F in campo inverso. A che cosa serve?** *(Luca Monticelli - Milano)*

• A niente! Col tasto CTRL è possibile, altresì, riprodurre altri simboli che generano effetti vari e già noti come quelli del controllo cursore. Tra l'altro, ti ricordo che col tasto "N" (che genera, appunto, una "N" in campo inverso) è possibile passare al modo maiuscolo minuscolo. Simula, in altre parole, il comando:
PRINT CHR$(14)

Sonoro che non suona

□ **Come posso controllare se il chip sonoro del mio computer è guasto dato che non riesco a fargli emettere alcun suono?** *(Baldassarre Di Silvestre - Gela)*

• Abbiamo più volte detto che per controllare la funzionalità della parte sonora di un computer è sufficiente digitare una delle decine di routine sonore pubblicate e, durante il

loro funzionamento, sintonizzare con calma il televisore con cui si è collegati. Se non si sente nulla i casi sono due: o il televisore è incompatibile col computer (può succedere) oppure il computer è guasto.

Col tuo Commodore 64 prova a far girare il programma che segue e che genera impulsi sonori continui:

```
100 POKE 54296,15:POKE
    54276,17
110 POKE 54277,9:POKE 5
    4278,0
120 POKE 54276,17:POKE
    54273,40
130 POKE 54272,38
140 FOR I=1 TO 100:NEXT
150 GOTO 100
```

Danni di passaggio da Vic 20 a C-64

☐ **Cosa può succedere e quali danni possono accadere se digito un programma scritto per il Vic 20 nel Commodore 64 e viceversa?** *(Andrea Armaroli - Bologna)*

• Assolutamente nulla di dannoso può accadere nei casi citati: al massimo il programma funziona male o provoca un "inchiodamento" della macchina. In quasto caso l'unico danno che ne deriva è quello relativo alla perdita del tempo impiegato per trscrivere il listato!

Simulazione, su Vic, del C-64

☐ **E' possibile ottenere col Vic 20, una mappa di memoria dello schermo come quello del C-64?** *(Davide Mazzeo)*

• Lo schermo del Vic 20 è formato da 22 colonne e 23 righe mentre quello del Commodore 64 da 25 e 40 colonne.

Il circuito integrato video del Vic è fatto in modo da gestire la zona di memoria dedicata allo schermo tenendo conto del formato citato. Non è quindi possibile modificare le dimensioni dello schermo del Vic in modo da portarlo alla grandezza del C-64.

Vi sono stati, nel passato, tentativi di commercializzare schede elettroniche che consentissero tali trasformazioni ma, a parte la pessima visualizzazione dei caratteri, avevano un costo pari a quello del computer stesso. E' ovvio che l'acquirente preferiva, fatti i debiti conti, acquistare direttamente un Commodore 64 che dava, tra l'altro, numerosi altri intuibili vantaggi.

Il LISP

☐ **Esiste il LISP per il Commodore 64?** *(Gian Franco Querciagrossa - Forlì).*

• Col Commodore 64 è possibile, almeno in teoria, implementare quasi tutti i linguaggi e sistemi operativi (compreso il CP/M grazie ad una scheda aggiuntiva). Non conosciamo, tuttavia, se sia disponibile anche il Lisp. Rivolgiamo pertanto la tua domanda ad eventuali Ditte che commercializzano tale prodotto.

I programmi gestionali

☐ **Gradirei vedere pubblicati programmi gestionali.** *(Sauro Gregnanin - Rovigo).*

• Purtroppo il numero di richieste in tal senso è troppo modesto per giustificare la pubblicazione di programmi gestionali. Articoli e listati di semplici routine finanziarie, al contrario, sono stati e saranno presi in considerazione per soddisfare sia gli studenti di ragioneria, sia coloro che, magari per semplice curiosità ed aggiornamento personale, intendono conoscere problematiche finanziarie.

Argomenti particolari o comunque complessi, verranno affrontati sull'altra nostra rivista, COMMODORE, destinata, appunto, agli utenti professionisti o "evoluti".

C-64 come Vu-Meter

☐ **Vorrei sapere se l'entrata Audio del C-64 è collegabile ad una sorgente sonora (ad esempio un amplificatore stereo) per ottenere disegni sul video, la cui esecuzione dipende dal segnale immesso (VU meter ecc.). Se ciò è ottenibile, quali sono le locazioni di memoria per attivarlo?** *(Marco Lucenti - Vittorio Veneto)*

• L'entrata audio del C-64 è collegata, tramite condensatore, direttamente ad un piedino del circuito integrato SID (6581) che gestisce il suono del computer. Il collegamento suddetto serve per creare effetti sonori speciali immettendo nel computer un'onda in grado di alterare il funzionamento stesso del SID.

La costruzione di un apparecchio in grado di svolgere la funzione di interfaccia tra computer e apparecchio esterno è possibile ma, come puoi intuire, richiede conoscenze di elettronica non indifferenti (digitalizzazione di un segnale analogico) ed esperienza di programmazione in Linguaggio Macchina data l'alta velocità di risposta necessaria.

In conclusione, un analizzatore di spettro, o un più modesto VU-Meter, richiede un'analisi di programmazione e la realizzazione di una serie di prototipi tra cui scegliere il più idoneo.

Problemi di sintonia

☐ **Il mio C-64, connesso col televisore, presenta righe trasversali fastidiose. Che cosa posso fare?** *(Giuseppe Pini - Firenze)*

• I casi sono due: il televisore che adoperi "slitta" in frequenza a causa della sua vetustà. Prova a collegare il computer con un apparecchio più recente e verifica che tutto funzioni a dovere.

Se il difetto persiste è probabile che il C-64 prevenga dal mercato inglese oppure che abbia perso l'allineamento. In questo caso devi inviare il calcolatore per la necessaria taratura.

Il Turbo non va su di giri

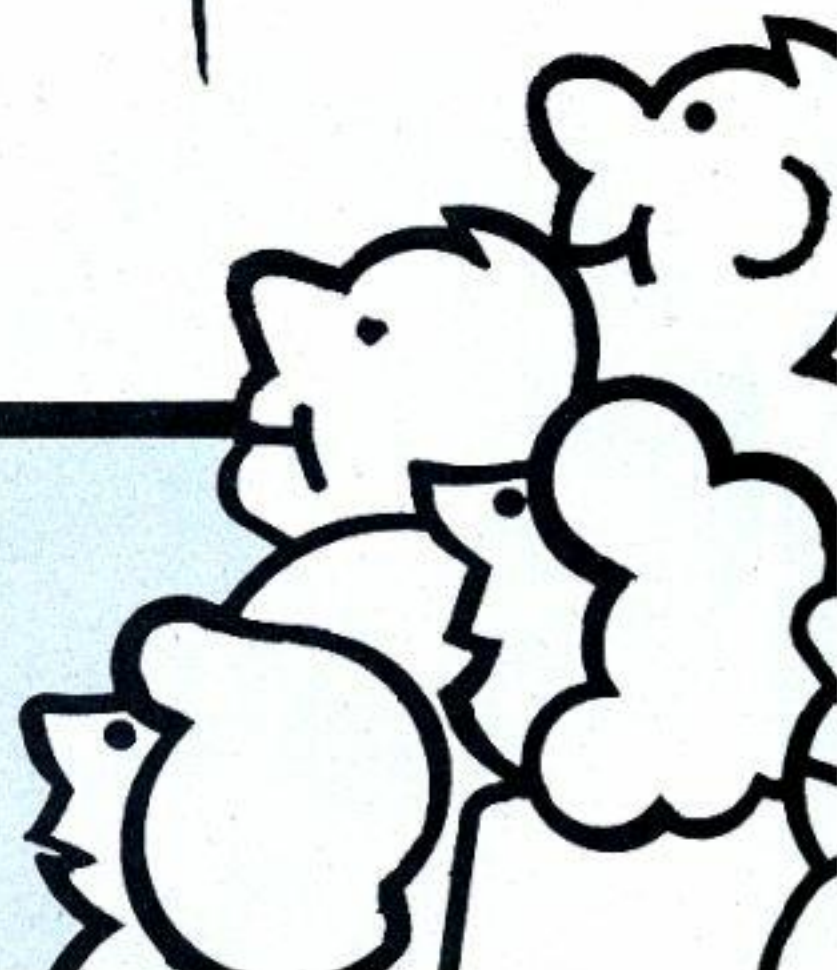
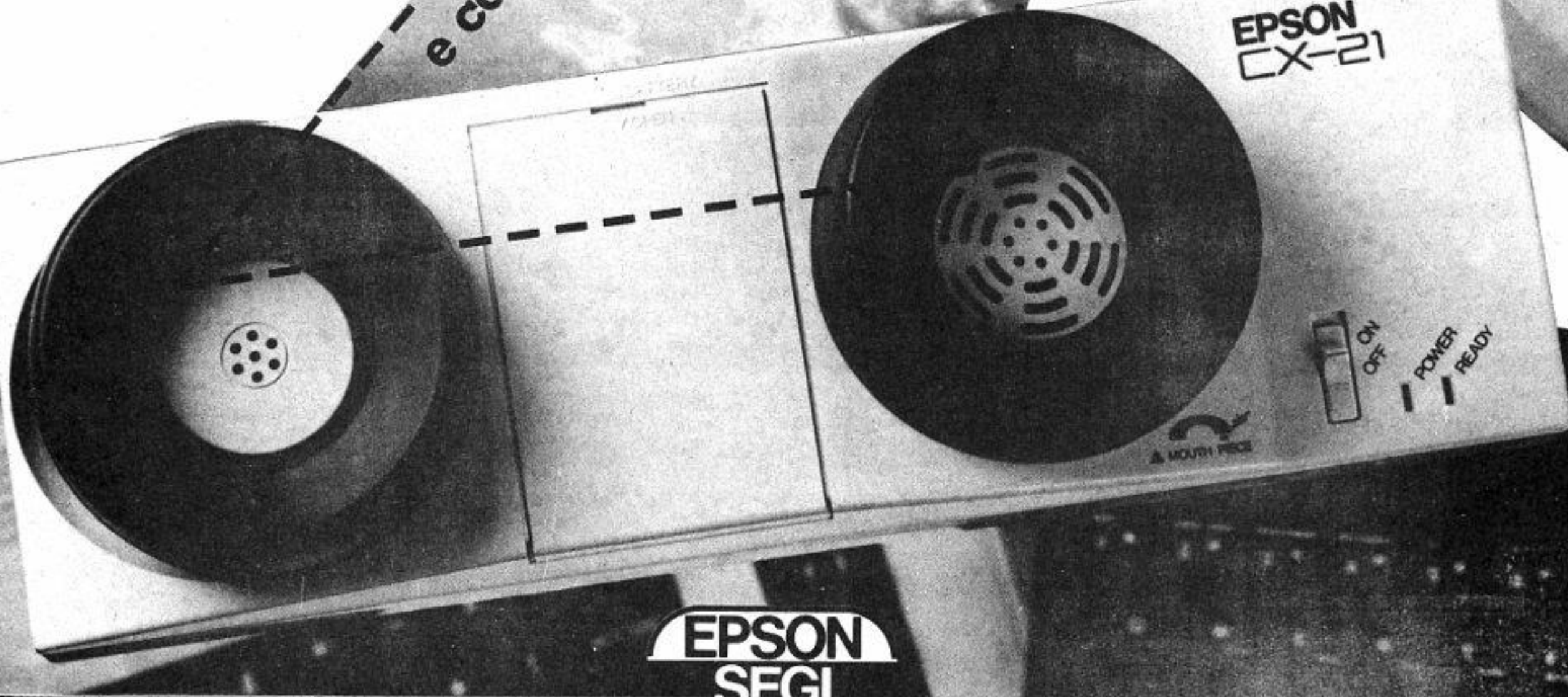
☐ **Il programma Turbotape, inserito nelle cassette della rivista Commodore Club, non gira col mio Vic 20. Come mai?** *(Tiziano Colletti - Albenga)*

• Il Turbotape è stato progettato per il Commodore 64 e, di conseguenza, non può essere adoperato sugli altri computer della Commodore. Ciò è dovuto al fatto che la cosidetta "gestione" della memoria e di numerose routine dei due computer sono piuttosto diverse tra loro.

C-64 americano

☐ **Purtroppo non sapevo che un computer acquistato in America non può funzionare in Italia ed ora mi trovo un apparecchio inutilizzabile. C'è qualcuno disposto ad effettuare le necessarie modifiche elettroniche?** *(Claudio Fadda - Firenze)*

• C'è qualche centro di riparazioni in grado di effettuare tali interventi? Fatecelo sapere, per favore. Molti sono gli utenti che, come il signor Fadda, si trovano involontariamente in difficoltà.

Alessandro de Simone

COMPUTERTELEFONA
I TUOI PROGRAMMI
CX-21 EPSON è un accessorio che dà un valore enorme al tuo calcolatore, quello dei terminali telecomunicazione.
CX-21 è un modem costruito con le tecnologie più avanzate, (modulatore - demodulatore) capace di convertire i segnali digitali in suoni che possono essere trasmessi con un normale telefono, in tutta sicurezza.
Con CX-21 puoi:
- "telefonare" i programmi direttamente al computer dei tuoi amici che posseggono un CX-21
- comunicare con personal computer di altre marche
- collegarti con le banche dati di tutto il mondo
CX-21 è progettato per adattarsi ai diversi apparecchi telefonici per consentire lo scambio veloce e facile di programmi e dati. CX-21 è costruito con speciali circuiti integrati CMOS, è piccolo, leggero e molto affidabile.
Caratteristiche
- Funzionamento a batterie ricaricabili (4 ore di autonomia)
- Funzionamento duplex o half duplex
- Velocità di trasmissione fino a 300 bit-sec.
- Interfaccia RS 232
- Dimensioni 297x95x42 mm
- Peso 700 gr.
Inoltre CX-21 può essere collegato a qualsiasi home o personal computer tramite interfaccia RS 232 - C.
e collegati alle banche dati di tutto il mondo
EPSON · SEGI Garanzia delle Quattro Stagioni
EPSON CX-21, il tuo computer in un mondo senza confini
EPSON CX-21
Accoppiatore acustico
EPSON CX-21
ON OFF
POWER
READY
MOUTH PIECE
EPSON SEGI

ORIALE · EDITORIALE

Il computer in vacanza

Molti sono ormai gli utenti che portano con sè le proprie apparecchiature nei luoghi di villeggiatura.

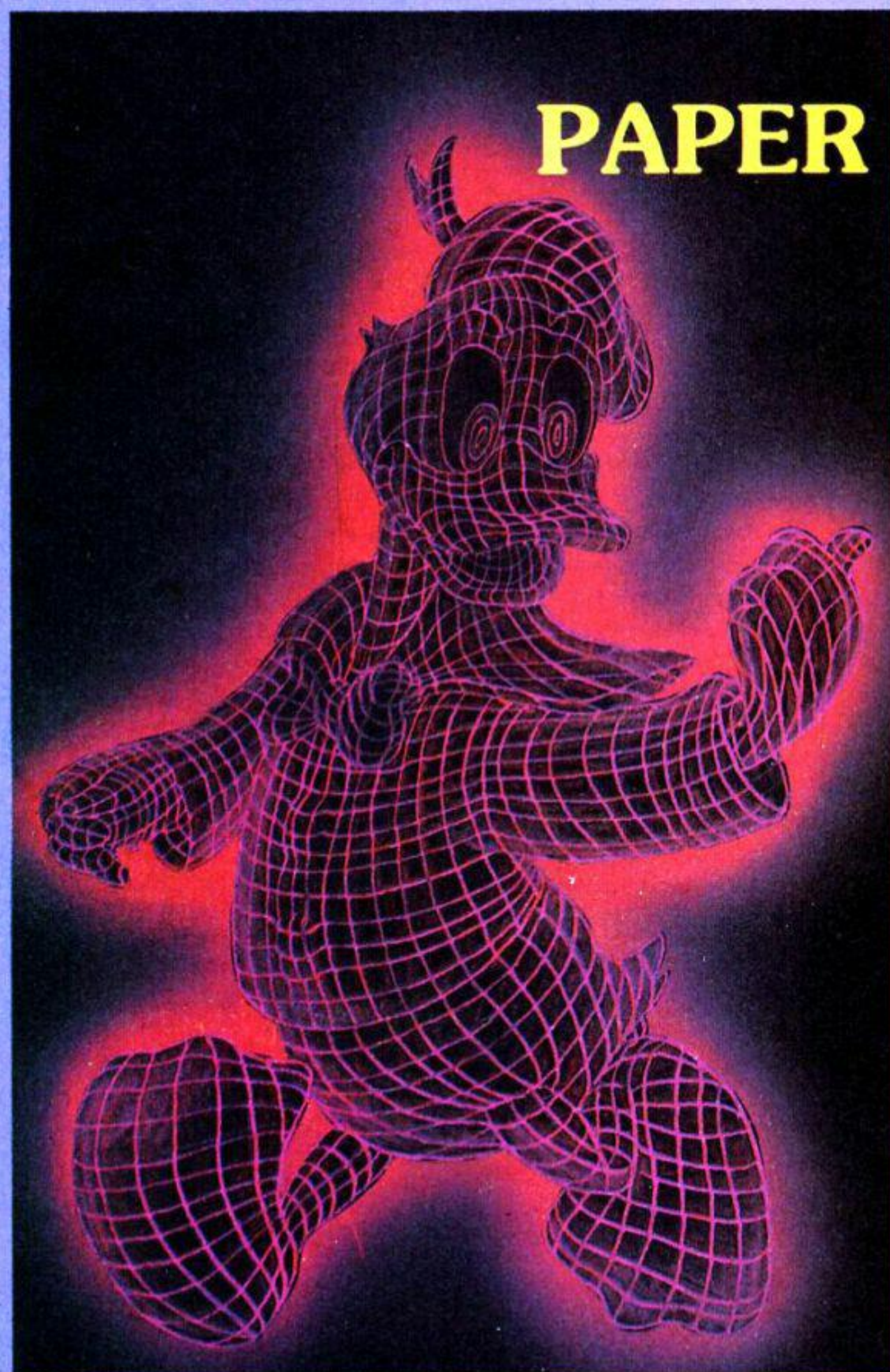
Si può trascorrere una giornata non proprio assolata, infatti, in compagnia di nuovi amici davanti allo schermo con un joystick in mano o un problema di scacchiera. E' probabile che anche l'albergatore, incuriosito, assapori i primi piacevoli momenti informatici e decida di acquistare un apparecchio da mettere magari a disposizione dei suoi ospiti.

Da parte nostra, oltre a proporre i consueti programmi originali e interessanti, pubblichiamo qualcosa di più "leggero", che si adatti al clima vacanziero di questi giorni: un quiz (mica tanto semplice), parole crociate (forse bizzarre), gli indovinelli dei programmi da una sola riga, ed altre cosette amene riempiono alcune delle pagine di questo numero, da leggere con tranquillità sotto l'ombrellone o seduti in pineta.

Agli altri argomenti più impegnativi penseremo più in là, dopo le ferie.

Per il momento sappiate che la scuola ed il lavoro, che riprenderemo tra un secolo, in settembre, potranno essere affrontati in maniera più sopportabile sapendo che a casa nostra ci attende il computer che, con un pò di fantasia, renderà più lunga questa dolce, pigra, lunga vacanza che ci stiamo godendo tutti...

Alessandro de Simone

Le immagini di questo fascicolo

PAPER TRON

Prima di entrare negli studios della Walt Disney Corporations, l'attore Jeff Bridges aveva l'abitudine di passare almeno un'ora davanti ai video game di casa sua. Obiettivo: entrare nello stato d'animo adatto a interpretare la sua parte. Un training psicologico comprensibile: protagonista in Iron, Jeff si trovava ogni giorno alle prese con colleghi ingaggiati nel ruolo di software e diaboliche ROM, tra bit amichevoli e ruscelli di energia, infine condannato a morte all'interno di un videogame.

Per realizzare la futuristica avventura del programmatore che entra nel computer, alla Walt Disney sono state mobilitate tutte le più sofisticate tecnologie elettroniche (del mondo reale). Uscito sugli schermi a fine 1982, Tron rappresenta ancora adesso la più originale e sofisticata opera nella storia della cinematografia mondiale.

Per ricostruire il mondo elettronico, la Walt Disney si è affidata a quattro tra le più importanti aziende di computer graphic: Mathematic Applications Group Inc. (Magi), Information International Inc. (Triple I), Robert Abel & Associates, Digital Effects. Per la maggior parte del tempo al lavoro su un set disadorno, con fondali neutri, le scene sono poi tutte passate attraverso i computer di queste aziende, con una postproduzione durata più di un anno.

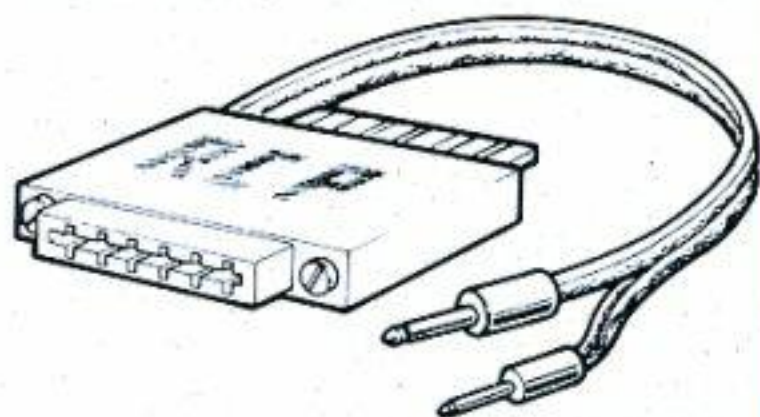
Alla Magi di New York, i calcoli necessari per ogni fotogrammi di film erano affidati a un sistema 3240 della Perkin Elmer (due Mbyte di memoria MOS e due drive da 80 Mbyte) interfacciato con un computer Celco CFR 4000, usato per generare su monitor le immagini.

Alla Triple I, main computer era un Foonley Fl, le figure erano affidate a un registratore PFR, mentre i programmatori usavano una tavola grafica da 40 per 60 pollici (un metro per un metro e mezzo). Quattro sistemi Hewlett Packard 9826 calcolavano il tempo di esposizione di ogni fotogramma, mentre un Cinetron 1100 B muoveva la telecamera.

Alla Walt Disney, un sistema Univac 98/80 indicava dove ogni fotogramma si trovava nelle 50 fasi del processo di produzione, dal layout al negativo. E il collegamento via terminale tra le aziende estere e la Walt Disney permetteva un risparmio di tempo per correzioni e aggiustamenti calcolabile tra i due e i cinque giorni pieni. Per farsi un'idea della mole di informazioni contenute in ogni scena, basta pensare che per ogni scena erano necessari dai 5 ai 75 milioni di calcoli, e che per formare un minuto di film ci volevano 1240 fotogrammi.

Moltiplicate, gente, moltiplicate. E, per cominciare, gustatevi il risultato dell'opera nelle immagini pubblicitarie in questo numero di Commodore Computer Club, per gentile concessione della Walt Disney Corporation.

ACCESSORI PER COMPUTER COMMODORE

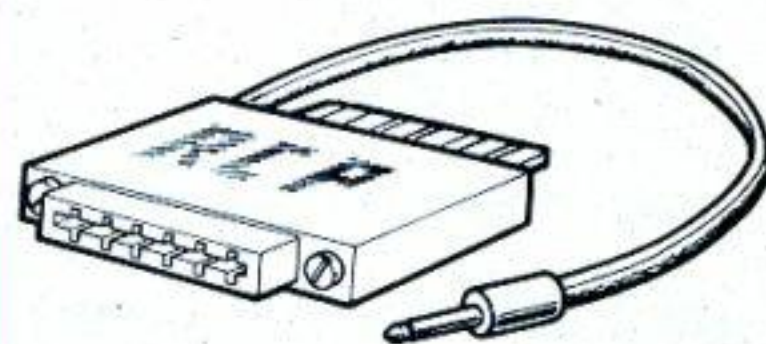
DUPLICATORE PROGRAMMI

Utile accessorio per fare copie tramite un registratore commodore e un registratore normale, di nastri protetti o con caricamento turbo.

Art. CD 102 **L. 30.000**

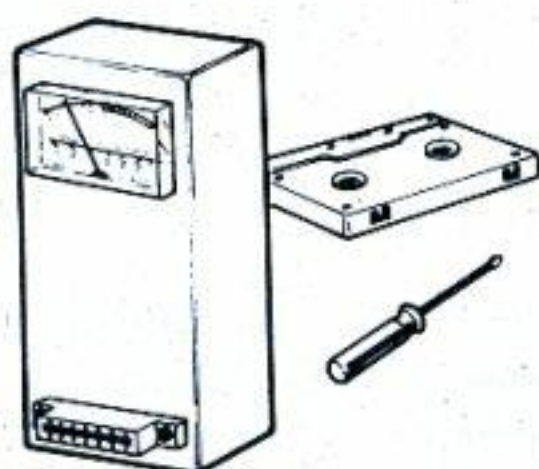
COPIATORE PROGRAMMI

Indispensabile accessorio per fare una copia, tramite due registratori commodore, di nastri protetti o con caricamento turbo.

Art. CD 103 **L. 30.000**

INTERFACCIA RICEZIONE RADIO

Indispensabile per registrare su registratore commodore i programmi speciali per computer trasmessi dalle emittenti radio.

Art. CD 104 **L. 30.000**

KIT ALLINEAMENTO TESTINE

Strumento indispensabile per la perfetta regolazione dell'AZIMUT nei registratori commodore o compatibili.

Art. CD 105 **L. 47.000**

RCP S.R.L. ELETTRONICA

Via Don Pasquino Borghi, 13
42017 NOVELLARA (Reggio E.)

Contenitore batteria tampone anti BLAK-OUT	Art. CD 117	L. 25.000
Interfaccia registratore	Art. CD 101	L. 30.000
Commutatore TV antenna/computer	Art. CD 108	L. 9.500
Tasto RESET per Vic 20 e C 64	Art. CD 109	L. 5.500
Cavo per TV con ingresso monitor	Art. CD 110	L. 18.000
Copritastiera in plexiglas	Art. CD 750	L. 18.000
Copritastiera in stoffa	Art. CD 755	L. 10.500
Borsa per trasporto C 64 e REGISTR.	Art. CD 760	L. 29.000
Vaschetta per 90 floppy D. con chiave	Art. CD 780	L. 45.000
Vaschetta per 40 floppy D. con chiave	Art. CD 770	L. 35.000
Tavola grafica SUPER SKETCH per C 64	Art. CD 790	L. 240.000
Conf. 10 dischetti ODP 1F 2D	Art. CD 700	L. 45.000
Conf. 10 dischetti KEY-DATA 1F 2D	Art. CD 710	L. 40.000
Conf. 10 dischetti COLORATI 1F 1D	Art. CD 730	L. 45.000
Conf. 10 dischetti HOBBY-FLOPPY 1F 1D	Art. CD 720	L. 29.500

ALIMENTATORE

Unità di alimentazione per C 64 e VIC 20 completo di interruttore con filtro di rete e soppressore di picchicontro i disturbi elettrici. Predisposto per batterie tampone anti BLAK-OUT (Art. CD 107).

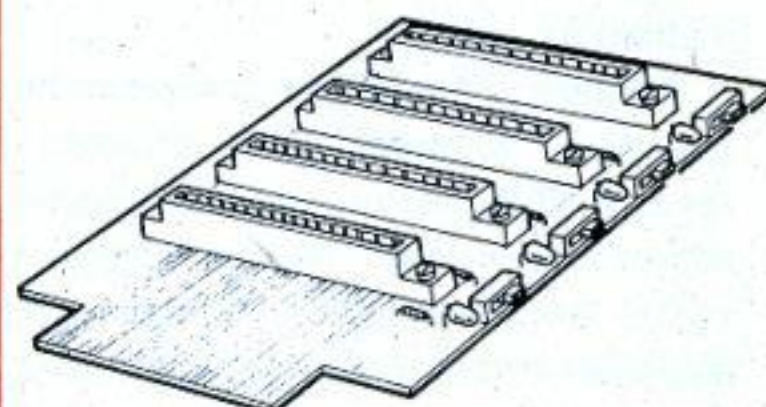
Art. CD 106 **L. 87.000**

BATTERIA TAMPONE ANTI BLAK-OUT

Consente il mantenimento dei dati in memoria nei computer in assenza di corrente elettrica per circa 30 minuti (da abbinare all'alimentatore Art. CD 106). Fornibile anche senza batterie (Art. CD 117).

Art. CD 107 **L. 85.000**

TUTTI I PREZZI SONO COMPRENSIVI DI IVA
NON SI ACCETTANO ORDINI INFERIORI A L. 30.000
CONTRIBUTO FISSO SPESE DI SPEDIZIONE L. 5.000

BUS A QUATTRO SLOT PER VIC 20

Amplia la possibilità della porta di espansione fino a 4 ingressi selezionabili di cui uno indirizzabile nell'area di memoria ROM, completo di tasto RESET.

Art. CD 100 **L. 55.000**

BUONO DI ORDINAZIONE

NOME - COGNOME

INDIRIZZO — N.

C.A.P. — CITTÀ — PROVINCIA

VOGLIATE INVIARMI IN CONTRASSEGNO

N.	Art.	L.
N.	Art.	L.
N.	Art.	L.
SPESE SPEDIZIONE		L. 5.000
PAGHERÒ AL POSTINO		L.

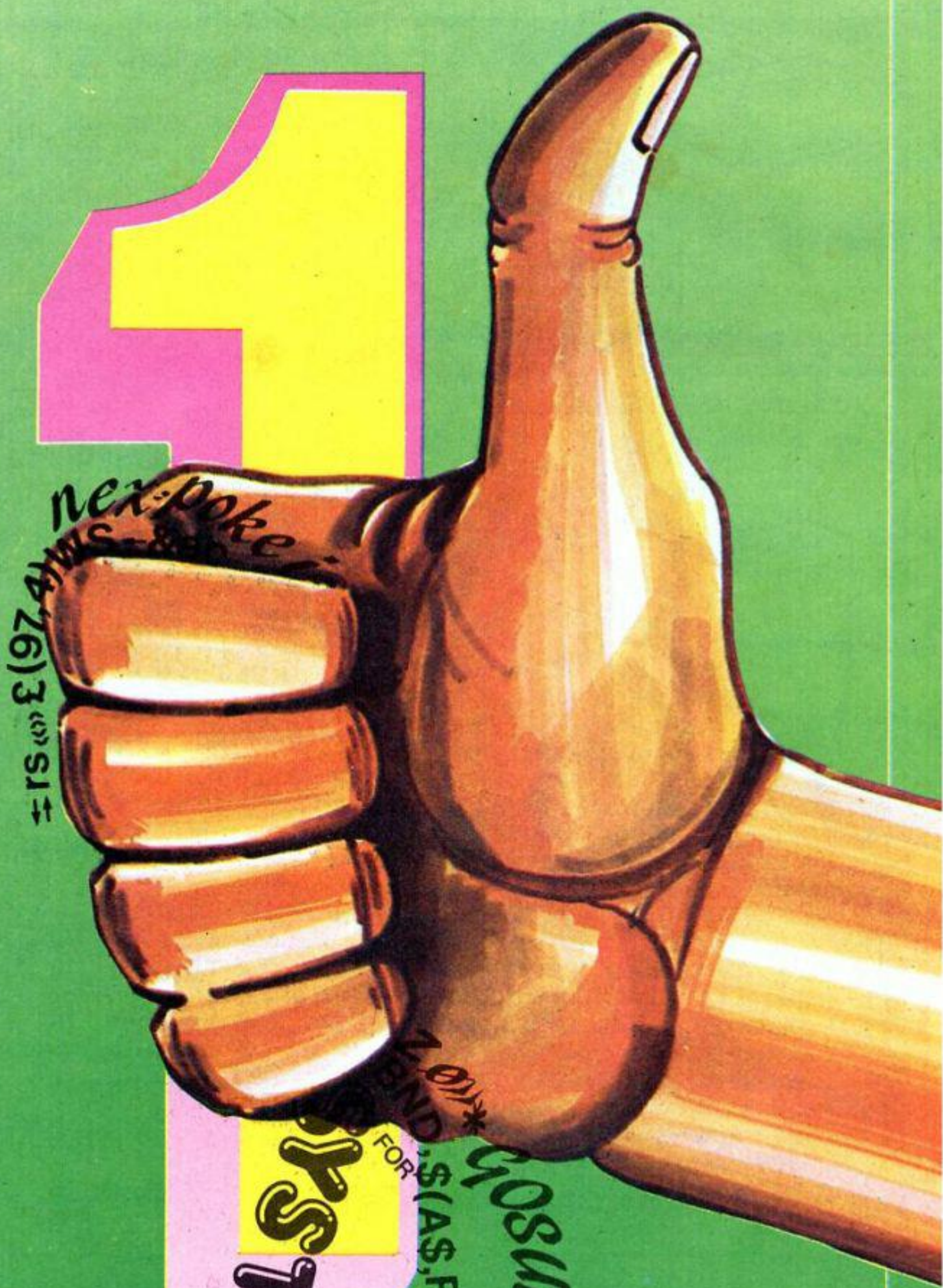
COMPUTER SERVICE VIA A. MANZONI, 49 - 42017 NOVELLARA (RE) - TEL. (0522) 661647

Questo mese la rubrica una riga si propone in una veste prettamente estiva.

Abitualmente ad ogni breve listato di questa rubrica, è abbinato il testo corrispondente. In questa occasione abbiamo deciso di incentivare l'interesse di voi lettori, apportando una significativa modifica, in ogni pagina sono poste da un lato le piccole routine e dall'altro i rispettivi testi messi in ordine sparso.

Starà a voi individuare i giusti accostamenti tra listati e relativi testi.

Questa operazione metterà a dura prova le vostre capacità di analisi di un programma e non sarà obbligatorio avvalersi di un computer per ultimare questa operazione. Se siete sotto un ombrellone o nella pace della montagna, muniti solamente della vostra rivista e di un blocco per appunti, potrete cimentarvi con questa simpatica sfida che vi proponiamo.

Lo spirito che anima la rivista Commodore Computer Club di questo mese, è la sua possibilità di utilizzo anche a prescindere dalla presenza della tastiera di un calcolatore.

1

Ciao. Una semplice e simpatica routine, che vi saluterà facendo ricorso al linguaggio macchina. *(La redazione)*

2

Tabellina. Questo programma simula una normale tabellina pitagorica. *(Ferrer)*

3

Miniscarabeo. Questo gioco necessita di almeno due concorrenti. Uno dei due dovrà fornire, quando gli verrà richiesto, una parola di suo gradimento. Ultimata questa operazione, l'altro partecipante al gioco vedrà visualizzate le tre prime lettere di questo vocabolo e in base a queste cercherà di indovinarlo. *(Cutolo)*

4

Sinusoidamente. Questa routine genera una sinusoide costituita unicamente da asterischi. *(Casartelli)*

5

Cursore magico. Potrete ottenere degli effetti magici tramite il cursore. *(Casartelli)*

6

Messaggi. Questa routine visualizza tutti i messaggi del linguaggio BASIC. *(Casartelli)*

7

Scacchiera. Questo breve programma creerà una scacchiera sul video del vostro Commodore. *(La redazione)*

8

Scalini. A cosa può servire un generatore di scalini? Probabilmente a nulla, però è simpatico da ossevarsi. *(La redazione)*

9

Critto/scritto. Dovete fornire un numero compreso tra 1 e 10 e poi la parola o la frase da crittografare.

Se poi vorrete decodificare la frase precedentemente introdotta, fornite il medesimo numero, ma con il segno negativo. *(La redazione)*

10

Cambia colore. Questa routine in linguaggio macchina farà mutare rapidamente i colori del bordo del Commodore 64. *(La redazione)*

A

```
2 INPUT A:INPUT A$:FOR X=1 TO LE
  N(A$):PRINTCHR$((ASC(MID$(A$,X
  ,1))+A) AND 255);:NEXT
```

B

```
1 DIM A(10,10):FOR I=1 TO 10:FOR
   J=I TO 10:A(I,J)=I*J:PRINTA(I
  ,J);:NEXTJ:PRINT:NEXTI
```

C

```
1 PRINTCHR$(14)+CHR$(147):FOR A=
  41118 TO 41767:S=PEEK(A):PRINT
  CHR$(S);:NEXT
```

D

```
1 FOR T=1 TO 360 STEP 20:L=SIN(T
  *π/180)*10+20:PRINT TAB(L)"●":
  NEXT:GOTO 1
```

E

```
1 FOR T=1 TO 4:READ S:PRINTCHR$(
  S):PRINT"[CLEAR]":NEXT:RESTORE
  :GOTO 1:DATA 172,187,190,188
```

F

```
1 INPUT "PAROLA";A$:PRINT"[CLEAR
  ]":PRINTLEFT$(A$,3):INPUT B$:I
  F A$=B$ THEN PRINT"OK":GOTO 1
```

G

```
1 PRINT"[RVS] ";:POKE 646,(A  AN
  D  1)+((TI/3E3)  AND  15):B=B+
  1+40*(B=39):A=A+1+(B=39):GOTO
  1
```

H

```
1 A=((N  OR  3)/4) AND 1:PRINTMI
  D$("[RIGHT][DOWN]",A+1,1)CHR$(
  175-5*A)"[LEFT]";:POKE 646,PEE
  K(161):N=N+1:GOTO 1
```

I

```
1 PRINT TAB(2)"CIAO";:POKE 900,2
  38:POKE 901,32:POKE 902,208:PO
  KE 903,96:SYS900:GOTO 1
```

J

```
1 POKE 49152,238:POKE 49153,32:P
  OKE 49154,208:POKE 49155,76:PO
  KE 49156,0:POKE 49157,192:SYS4
  9152
```

11

Reverse. Tramite questa routine, potrete porre in reverse lo schermo del vostro video. *(La redazione)*

12

Traccia circonferenze. Non è proprio Giotto, ma pensando che il tuo computer riesce a fare ciò con solo una riga di programma, ogni imperfezione è scusata. *(Francesco Serci)*

13

Curiosità. Digitate questa una riga e poi battete il comando LIST .... otterrete uno strano effetto. *(La redazione)*

14

Capovolgi. Questa routine inverte una stringa fornita in input dall'utente. *(La redazione)*

15

Variabili. Questa breve routine si può rivelare utilissima, ponendola al termine di un programma da voi utilizzato, vi segnalerà tutte le variabili utilizzate. *(La redazione)*

16

Fondale. La routine in linguaggio macchina muta i colori del fondale del vostro Commodore 64. *(La redazione)*

17

Coriandoli. Una cascata di coriandoli sullo schermgidel tuo computer. *(La redazione)*

18

Stampa casuale. Mediante l'istruzione RND, contenuta in questa breve routine, potrete stampare dei caratteri in maniera casuale. *(Bernardo Nuccelli)*

19

Cursore. Potrete decidere la velocità di spostamento del vostro cursore, tramite questa semplice una riga *(La redazione)*

K

```
1 POKE 4096,238:POKE 4097,33:POK
  E 4098,208:POKE 4099,76:POKE 4
  100,0:POKE 4101,16:SYS4096
```

L

```
1 INPUT "[CLEAR]";X,Y:FOR A=1 TO
   85 STEP .05:POKE 1024+20+COS(
  A)*X+INT(12+SIN(A)*Y)*40,81:NE
  XT
```

M

```
1 REM "██████CIAO
```

N

```
1 FOR A=0 TO 15:POKE 646,A:B=INT
  (RND(TI)*200):PRINT TAB(B)"●":
  NEXT:GOTO 1
```

O

```
1 A=PEEK(45)+PEEK(46)*256:B=PEEK
  (47)+PEEK(48)*256:FOR X=A TO B
   STEP 7:PRINTCHR$(PEEK(X));CHR
  $(PEEK(X+1)):NEXT
```

P

```
1 FOR X=1024 TO 2023:POKE X,(( N
  OT (PEEK(X)) AND 128) OR (127
  AND  PEEK(X))):NEXT
```

Q

```
1 INPUT A$:FOR X=0 TO LEN(A$)-1:
  B$=B$+MID$(A$,LEN(A$)-X,1):NEX
  T:PRINTB$
```

R

```
1 A=INT(RND(1)*1001):B=INT(RND(1
  )*17):POKE (1023+A),102:POKE (
  55296+A),B:GOTO 1
```

S

```
1 INPUT "VELOCITA'";A:POKE 56325
  ,A
```

20

Schedina. Provate a fare tredici al totocalcio, mediante questa piccola routine. *(La redazione)*

21

Macchina. Una sivpaticaiautovellura percorrerà lo scharmo del vostro Commodore 64. *(La redazione)*

22

9000 byte. Volete espandere la memoria disponibile sul Commodore 64 a 90000 byte? Se lo desiderate dovete porre in esecuzione questa routine quando è visualizzato sul vostro televisore il numero di byte liberi disponibili. Il vostro ingegno potrà far ulteriormente aumentare la quantità di memoria disponibile. *(La redazione)*

23

Lucciola. Una lucciola si muove con andamento casuale sul video del vostro televisore. *(La redazione)*

24

Esami. Questa breve routine, vi consente di conoscere la vostra media universitaria dopo che avrete sostenuto il prossimo esame. Vi verrà richiesto il numero degli esami sostenuti e la somma complessIva dei voti conseguiti. In base a questi 2 dati, vi verrà fornita la media a secondo del voto che riporterete nel prossimo appello. *(Manna)*

25

Moschino. Un moschino si muoverà pazzamente sul monitor del vostro televisore. *(La redazione)*

26

Mare. Prima di porre in esecuzione questa breve routine, vi consigliamo di ingerire delle pillole contro il mal di mare. *(Bevilacqua)*

27

Media aritmetica. L'una riga in questione, consente di ricavare la media aritmetica di una serie di dati impostati dall'utente. *(La redazione)*

28

Interrogate. Vi offriamo un mini monitor, che consente di visualizzare il contenuto delle locazioni di memoria comprese tra due specifici indirizzi.

Inoltre verrà evidenziato il valore ASCII di queste locazioni. *(La redazione)*

T

```
1 INPUT A,B:FOR K=A TO B:N=PEEK(
  K):PRINTK;N;CHR$(N):PRINT:FOR
  P=1 TO 100:NEXTP:NEXTK:GOTO 1
```

U

```
1 A$=" ┌─┐─ ":B$="[DOWN][5 LEFT]'
  O—O'":PRINT"[CLEAR]" TAB(A)A$B
  $:IF A<255 THEN A=A+1:GOTO 1
```

V

```
1 PRINT"[HOME][3 DOWN][17 RIGHT]
  30000"
```

W

```
1 A$="_--‾--_":FOR I=0 TO 97:A$=
  MID$(A$,7,1)+MID$(A$,1,6)+MID$
  (A$,1,6):PRINTA$;A$;A$;A$;A$:N
```

X

```
1 PRINT"[CLEAR]":PRINT"ESAMI";:I
  NPUT A:PRINT"VOTO";:INPUT B:FO
  R I=18 TO 30:PRINTI":"(I+B)/(A
  +1):NEXT
```

Y

```
1 A$="[UP][RIGHT][DOWN][LEFT]":P
  RINT"[LEFT][RVS] "MID$(A$,RND(
  1)*4+1,1)"[LEFT][RVS]*";:FOR I
  =1 TO 30:NEXT:GOTO 1
```

Z

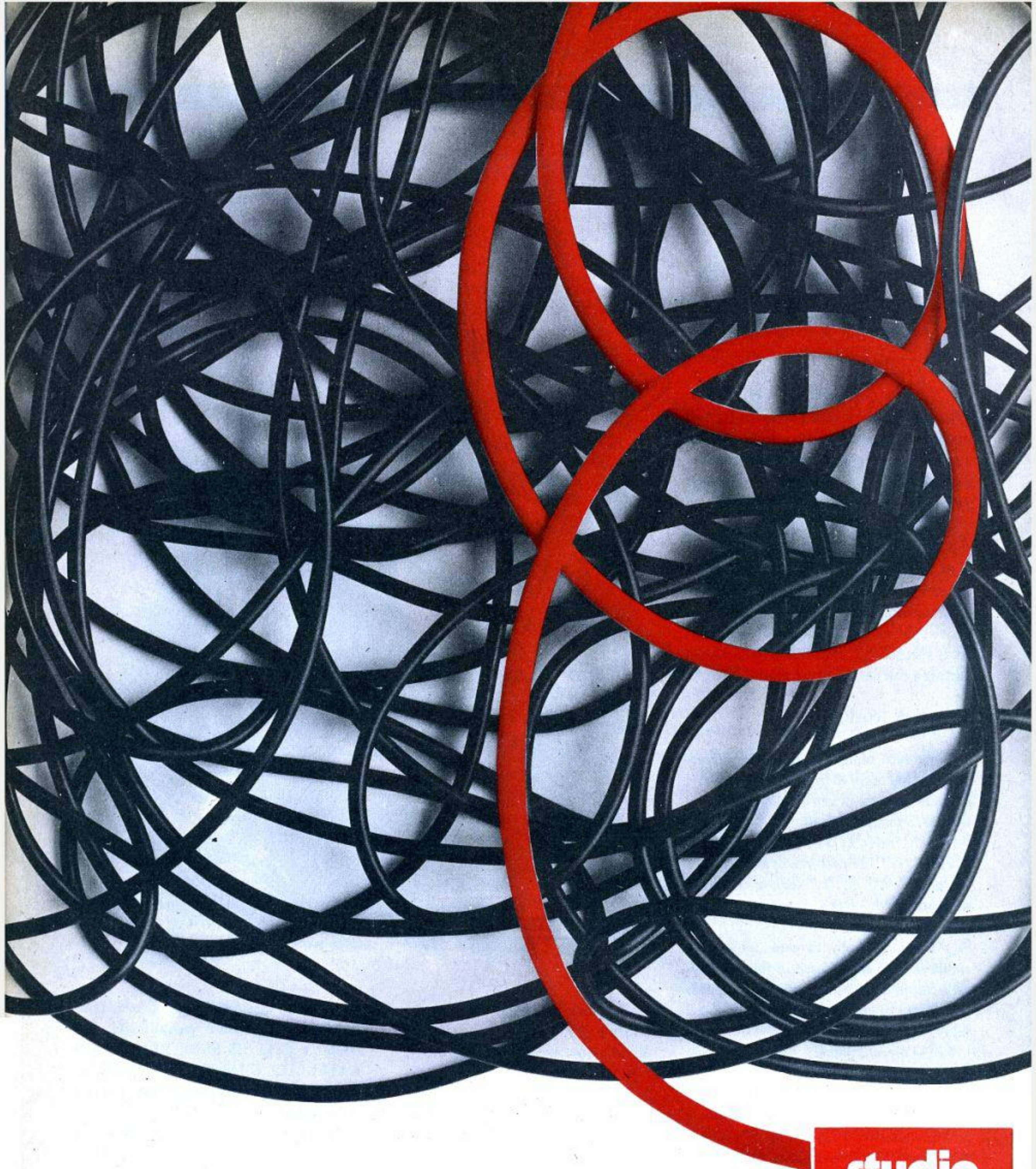
```
1 A=INT(RND(1)*1000):B=1024:POKE
   B+A,81:FOR T=1 TO 200:NEXT:PR
  INT"[CLEAR]":GOTO 1
```

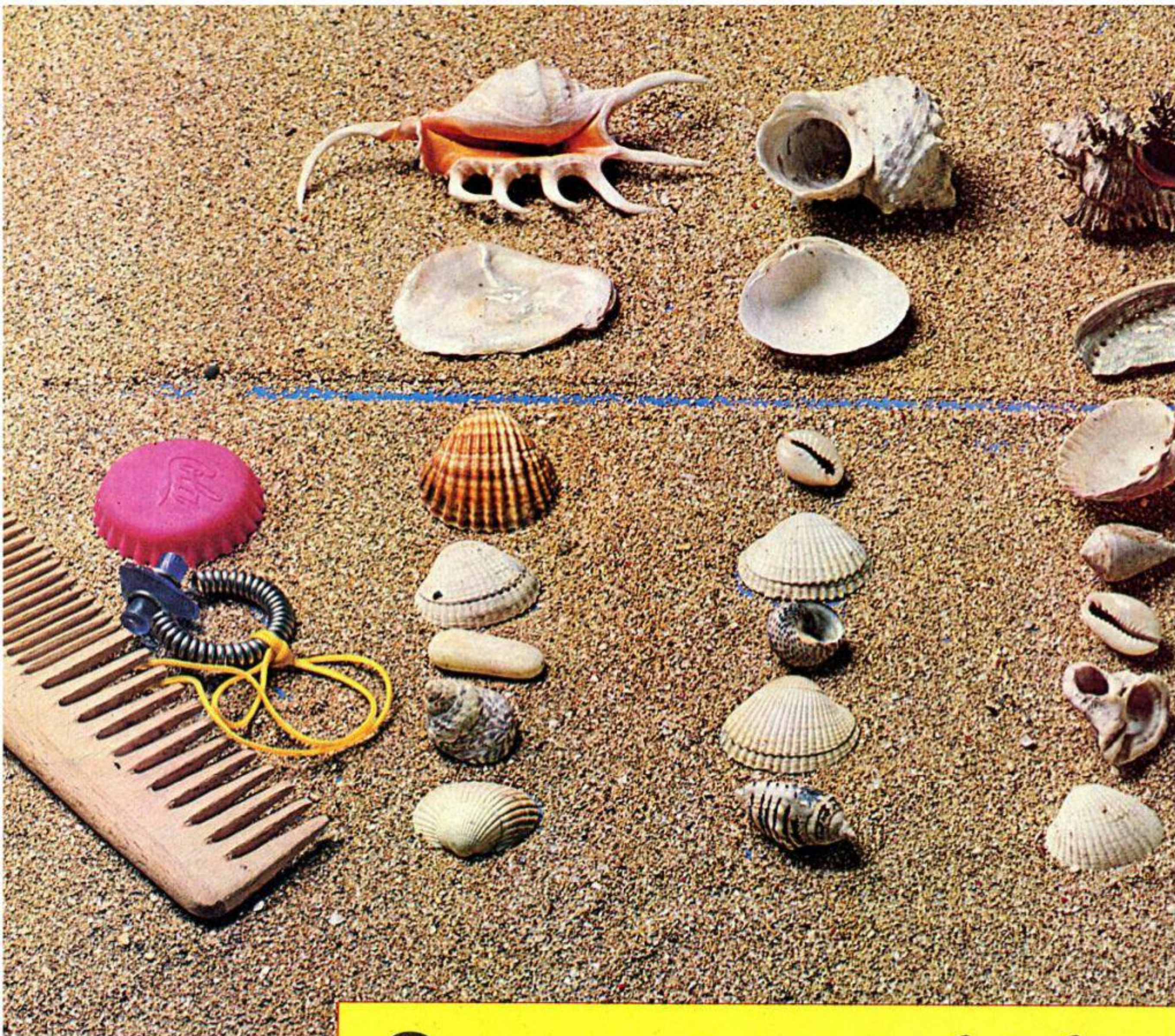
&

```
1 INPUT "QUANTI NUMERI";N:FOR I=
  1 TO N:PRINT"NUMERO"I;:INPUT V
  :S=S+V:NEXT:PRINT"LA MEDIA E'[
  RVS]"S/N
```

£

```
1 A$(0)="X":A$(1)="1":A$(2)="2":
  PRINT"[CLEAR]":FOR I=1 TO 13:A
  =RND(1)*3:PRINTA$(A):NEXT
```

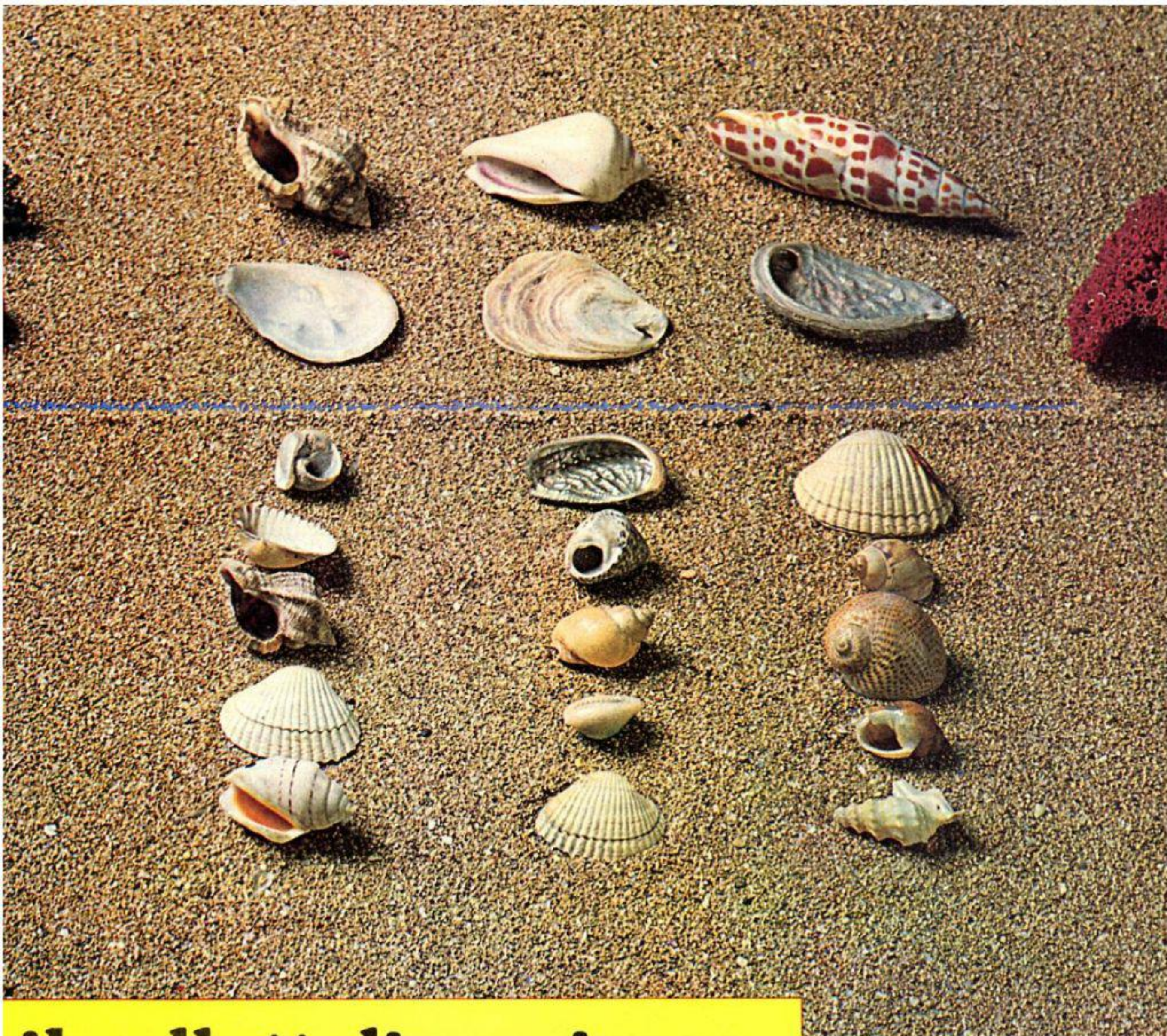
Gli accoppiamenti sono pubblicati a pagina. 51

SPE

Cari amici, questo numero
di luglio/agosto
di Commodore Computer Club è stato
da noi dedicato quasi completamente
ad allietare le vostre vacanze; che siate
al lago, in montagna, al mare o chissà
dove altro, siamo sicuri che non vi
sarete fatti scappare l'amata rivista in
edicola e che adesso vi accingete con
ansia a sfogliarne le pagine; ebbene,
tutte quelle che seguono, contengono
articoli da leggere in tutta rilassatezza
(sulla spiaggia, per esempio); potrete
poi approfondirli con calma davanti
alla tastiera e sassi e conchiglie,
oppure un robot fatto di scatole di
fiammiferi che imparano a giocare, le
parole crociate a tante altre cose
ancora. Buon divertimento!
CIALE VACANZE

Con sassi e conchiglie

Più veloce di molte calcolatrici elettroniche, l'abaco è il più diffuso utensile di calcolo di tutto l'est asiatico. Imparare ad usarlo non è difficile: basta un po' di buona volontà ed un pizzico di pazienza orientale. Ma noi vi proponiamo qualcosa di più. Certi che passerete la maggior parte delle vostre vacanze mollemente sdraiati su qualche spiaggia, mettete a frutto il vostro tempo: con dei sassi, o delle conchiglie, o entrambi, simulate un abaco ed imparate ad usarlo.

Tornati in città, fate il diagramma di flusso e preparate un programma per simulare su video il funzionamento del pallottoliere. Il programma migliore sarà pubblicato su Commodore Computer Club ed al suo autore verrà offerto in regalo... indovinate cosa? Un pallottoliere cinese in carne ed ossa!

RASSOMIGLIA al pallottoliere col quale da bambini abbiamo imparato a contare e lo si ritrova in numerosi paesi dell'Est, dalla Russia al Giappone.

L'abaco rappresenta l'"apparecchiatura" più diffusa negli uffici e nelle aziende cinesi. In tutta la Repubblica Popolare non c'è ufficio od abitazione che ne sia sprovvista e il suo uso è obbligatorio nelle scuole elementari e commerciali. Ovunque vada, il cinese lo porta con sè e spesso

il pallottoliere cinese

non riesce a contare senza questo strumento, la cui utilizzazione è semplice e fa concorrenza, dal punto di vista della velocità di calcolo, a quasi tutte le calcolatrici elettromeccaniche.

Approccio

L'abaco si divide in due zone: una superiore con due palline che funzionano da contatori per le cinquine ed una inferiore con cinque contatori per le unità, le decine, le centinaia, le migliaia, i milioni, i miliardi, ecc..

Per impostare un numero solitamente si usa: a) l'indice, b) il pollice e l'indice, oppure c) il pollice, l'indice ed il medio. Il sistema migliore sembra il secondo.

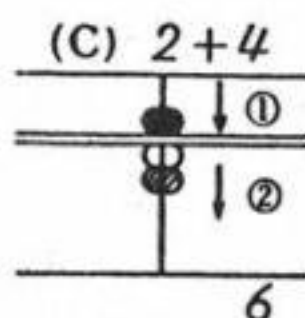
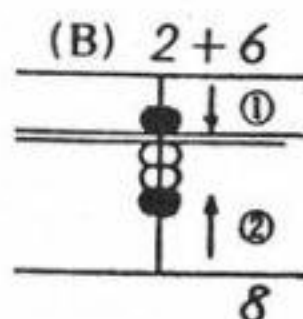
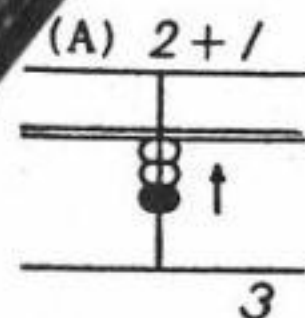
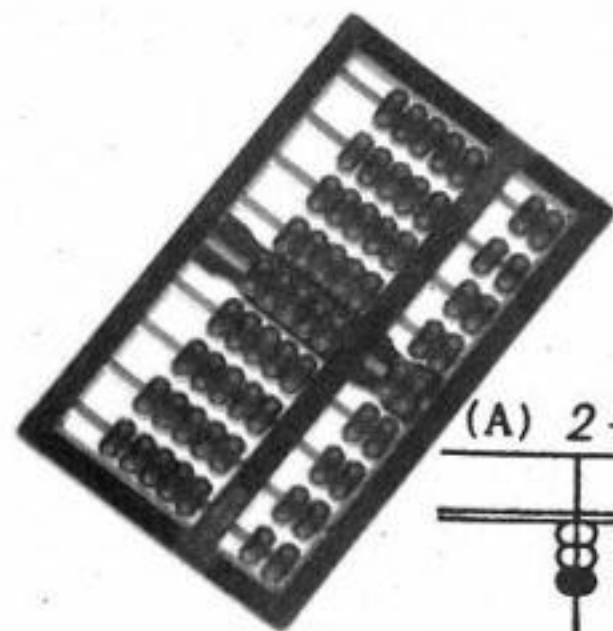
Il pollice serve per "aggiungere" i contatori nella zona inferiore (ossia spingerli verso la linea di demarcazione); l'indice per: a) toglierli, nella zona inferiore, b) aggiungerli o toglierli, in quella superiore. In particolare le operazioni effettuabili sono:

1/ aggiungere uno o più contatori;

2/ toglierli;

3/ aggiungere un contatore di cinquina e sostituire (togliere) uno o più contatori d'unità;

4/ aggiungere un contatore d'unità e toglierne uno di cinquina;

5/ togliere un contatore dalle unità e posizionarlo al posto delle

decine (passandolo, cioè, nella colonnina immediatamente più a sinistra);

6/ passare un contatore dalla posizione di decina a quella delle unità.

L'addizione

Spieghiamoci con degli esempi.

• *2 + 1.* Nell'esempio prima si posizionano 2 contatori, quindi si aggiunge una terza unità col pollice.

• *2 + 6.* In questo caso, si imposta prima il 2 col pollice; poi si posiziona con l'indice una cinquina nella parte superiore e col pollice una unità nella zona inferiore (5 + 1 = 6). Il risultato è un contatore di cinquina e tre unità (5 + 3 = 8).

• *2 + 4.* Dapprima s'imposta il 2, (posizionando col pollice due contatori d'unità), quindi si aggiunge il quattro. Poichè nella colonna non sono disponibili altri 4 contatori d'unità, si procede aggiungendo un contatore di cinquina e togliendone in basso uno d'unità (5 - 1 = 4). Sulla colonnina resterà una cinquina più una unità, ossia 6 che è il risultato ricercato.

• *9 + 7.* S'imposta il 9 (una cinquina e 4 unità), quindi si aggiunge il 7. Questo, non essendo disponibile nella stessa colonna è digitato con un contatore di decina nella posizione immediatamente a sinistra, meno tre unità, sottratte al 9 già impostato (10 - 3 = 7). Sull'abaco rimarrà il risultato 16.

• *36+75.* Per sommare numeri a due cifre si opera da sinistra verso destra, cioè dalle decine alle unità. In questo caso, prima si aggiungono le 7 decine di 75 operando come nell'esempio d, quindi si aggiungono le 5 unità sempre con lo stesso metodo.

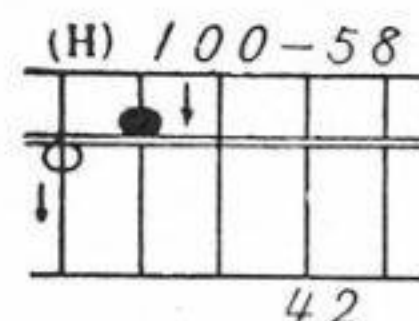
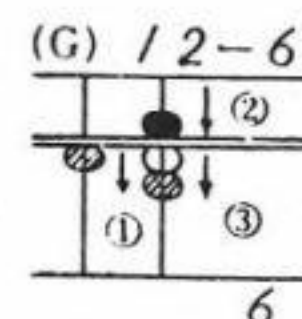
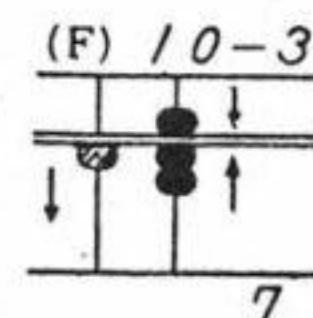
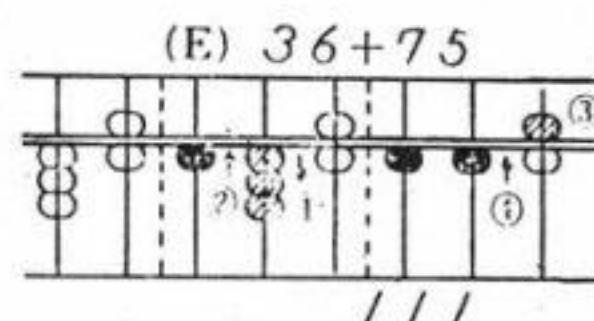
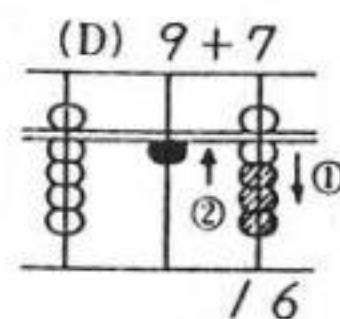
Sottrazioni

• *10 - 3.* In questo caso non abbiamo contatori d'unità, per cui si procede togliendo il contatore di decina e sottraendo da esso il 3 espresso nella forma -10 + 3. Per l'occidentale l'operazione sembra superflua, ma evidentemente le operazioni semplicissime vanno fatte mentalmente e visualizzate con l'abaco.

• *12 - 6.* Come nel caso precedente nella colonnina delle unità non abbiamo unità sufficienti per sottrarre il 6, che quindi viene impostato come -6 = -10 + 4, ossia togliendo il contatore di decina ed aggiungendo 4 unità. Poichè nella colonnina non sono disponibili, queste vengono aggiunte con un contatore di cinquina meno uno d'unità. L'operazione, in termini algebrici, si svolge nel modo seguente:

12 - 6 = 12 - (10-4) = 12 - -|10 - (5-1)| = 6

• *100 - 58.* Operando con numeri a due o tre cifre, si proce-

de sempre dalla sinistra verso la destra. In questo caso, prima si sottrae il 5 di 58 dalle centinaia (5 = 10 - 5), quindi si sottrae l'8 (-8 = -10 + 2).

Un esercizio che ha impegnato tutte le generazioni di studenti cinesi consiste nel sommare nove volte il numero 123.456.789. Il risultato corretto è 1.111.111.101 dal quale successivamente si sottrae nove volte consecutive 123.456.289 per ritornare al risultato "0". Se superate questo test potete passare alle operazioni successive.

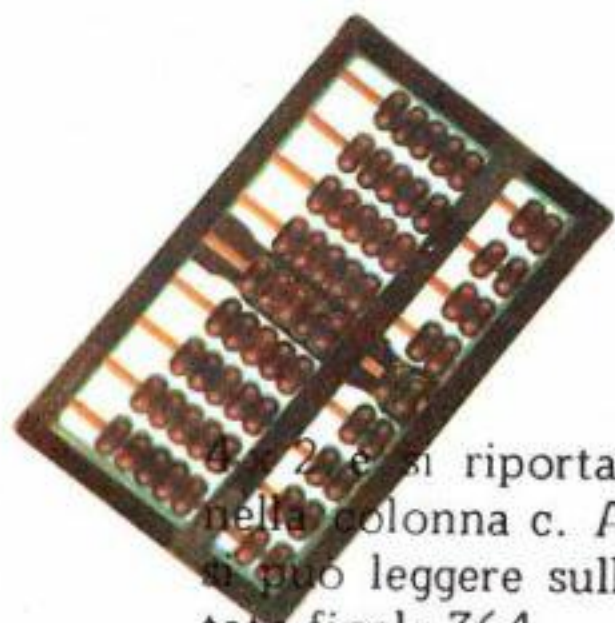
Moltiplicazione

Cominciamo dalla determinazione della posizione della cifra delle unità nel prodotto.
1/ Nel caso in cui il moltiplicatore è un numero intero o misto (intero e decimale), bisogna prima contare le cifre della parte intera, quindi ci si sposta d'un numero uguale di posizioni sulla destra della cifra delle unità del moltiplicando; l'ultima posizione trovata corrisponde a quella delle unità del prodotto da calcolare.
2/ Se il moltiplicatore è un decimale e tra la virgola e la prima cifra significativa abbiamo degli zeri si contano questi ultimi, quindi ci si sposta di un egual numero di posizioni sulla sinistra del moltiplicando fino a determinare la colonnina delle unità del prodotto.
3/ Se, infine, il moltiplicatore è un decimale e non c'è alcuno zero tra la virgola e le cifre significative dell'espressione, la posizione delle unità del prodotto corrisponde con la colonnina delle unità del moltiplicando.

Vediamo ora con degli esempi come si svolge l'operazione.
• *67 x 2.* Comparate l'ultima cifra del moltiplicando (il 7 della colonnina b dell'esempio) con il moltiplicatore 2.

2 x 7 dà 14, quindi, dopo aver tolto 7 dalla posizione b si mette al suo posto 1 – ossia la prima cifra di 14 – e 4 nella colonnina alla sua destra (c). Successivamente si passa all'altra cifra del moltiplicando (il 6 di 67) e, poiché 2 x 6 dà 12, si annulla il 6 della colonnina a e in sua vece si posiziona l'1 di 12; il 2 di 12 va posizionato nella colonnina b, cioè va addizionato alla cifra che vi troviamo. Sull'abaco, leggeremo il prodotto 134.
• *9 x 7.* In questo caso, sapendo già che 9 x 7 = 63, si posizionerà prima il 6 al posto del 9 ed alla sua destra si metteranno i tre contatori d'unità per il 3.
• *26 x 14.* Nella figura corrispondente vediamo che il moltiplicatore è un integrale a due cifre, quindi ci spostiamo di due posizioni sulla destra delle unità del moltiplicando (colonna d).

Procedimento:
1/ Calcoliamo il prodotto dell'ultima cifra del moltiplicando (col. b) con la prima del moltiplicatore (col. 1);
1 x 6 dà 6, quindi toglieremo il 6 dalla b e posizioneremo il prodotto sulla c;
2/ si calcola con lo stesso procedimento il prodotto della seconda cifra del moltiplicatore (il 4 di 14) e poiché 4 x 6 dà 24 si aggiunge il 2 di 24 nella colonnina c (dove già abbiamo un 6) e il 4 di 24 sul contatore immediatamente più a destra (colonna d);
3/ ora si riporta il risultato 2 di 1 x 2 nella colonna b, dopo aver tolto il 2 del moltiplicando dalla a. Infine si esegue mentalmente

Visualizzata nelle sue linee di progetto grafico, ecco ancora una delle scene del film Tron.

4 x 2 e si riporta il prodotto 8 nella colonna c. A questo punto si può leggere sull'abaco il risultato finale 364.

• *678 x 345.* Cominciamo con la determinazione della colonnina delle unità del prodotto. Il moltiplicatore è un numero intero a tre cifre, quindi contiamo tre posizioni sulla destra a partire dalla colonnina delle unità del moltiplicando: la posizione delle unità del prodotto corrisponde alla colonna f.

Il seguente diagramma mostra i diversi passi di calcolo da eseguire sull'abaco:

```
    2 4         (1)
      3 2       (2)
        4 0     (3)
  2 1           (4)
    2 8         (5)
      3 5       (6)
1 8             (7)
  2 4           (8)
    3 0         (8)
-----------
2 3 3 9 1 0
```

Ecco un altro esempio. Vogliamo moltiplicare 4,076x0,028 e come prima cosa determiniamo la colonna delle unità del prodotto.

to. Come si vede, il moltiplicando ha uno 0 dopo la virgola e la prima cifra significativa decimale; pertanto, conteremo una posizione a sinistra della colonnina delle unità del moltiplicatore (b), ossia stabiliremo che le unità del prodotto vanno posizionate in a. Ecco ora, la sequenza dei passi di calcolo (indicati dal numerino cerchiato):

• Dopo esservi esercitati posizionando nell'abaco sia il moltiplicando che il moltiplicatore, provate ad eseguire altri esercizi, posizionando solo l'uno o l'altro. I cinesi solitamente li tengono entrambi in memoria, allo scopo di rendere il calcolo più veloce.

Divisione

Anche in questo caso, la prima cosa da fare è determinare la posizione delle unità del quoziente.

1/ Si contano le cifre della parte intera del divisore e ci si sposta d'un egual numero di posti + 1 verso la sinistra della colonnina delle unità del dividendo determinando, così, dove posizionare le unità del quoziente.

2/ Se tra la virgola e la prima cifra significativa decimale ci sono degli zeri, ci si sposta verso la destra della colonnina delle unità del dividendo d'un egual numero di posizioni meno una, quanti sono gli zeri tra la virgola e la prima cifra significativa decimale. Se, invece, tra la virgola e la prima cifra tra la virgola e la prima cifra decimale significativa c'è un solo zero, la colonnina delle unità del quoziente coinciderà con le unità del dividendo.

3/ Se tra virgola e cifra decimale significativa non ci sono zeri, le unità del quoziente saranno posizionate nella prima colonnina a sinistra delle unità del dividendo.

Ecco ora alcuni esempi di calcolo.

• *93 : 3.* In questo caso il divisore è un numero intero ad una cifra, quindi ci sposteremo sulla sua sinistra di una posizione + 1, dove segneremo le unità del quoziente (B).

Il procedimento per il calcolo vero e proprio si svolgerà in questo modo:

1/ Componiamo la prima cifra del dividendo (il 9 della colonnina C) con il divisore 3 e, sapendo che il 3 nel 9 è contenuto 3 volte, posizioniamo il risultato 3 nella colonnina A.

2/ Moltiplichiamo il divisore 3 col primo risultato posizionato nella colonna delle unità A del quoziente (3). Il risultato 9 va sottratto al 9 della colonnina C.

3/ Ripetiamo l'operazione con la seconda cifra del dividendo (il 3 della colonnina D) e, sapendo che il 3 nel 3 è contenuto 1 volta, riportiamo l'1 nella B.

4/ Moltiplichiamo il divisore 3 per la seconda cifra del quoziente 1, quindi sottraiamo il risultato dalla colonnina D. A questo punto leggeremo il risultato 31.

• *1476 : 12.* In questo esempio il divisore è un numero intero a due cifre; quindi ci sposteremo di 2 + 1 posizioni a sinistra delle unità del dividendo (colonnina F) dove segneremo le unità del quoziente. I passi di calcolo sono i seguenti:

1/ Dividiamo la prima cifra del dividendo (l'1 di 1476) per la prima del divisore (l'1 di 12) e sapendo che 1 è contenuto nell'1 una sola volta, segnamo 1 nella colonnina A, quale prima cifra del quoziente.

2/ Moltiplichiamo l'1 di A con la prima cifra del divisore (cioè l'1 di 1476). 1 x 1 = 1 che sottrarremo dalla colonna C.

3/ Moltiplichiamo il 2 della seconda cifra del divisore 12 per la prima cifra del quoziente e sottrarremo il risultato (1 x 2 = 2) dalla D ottenendo un nuovo dividendo (276).

4/ Ripetiamo le operazioni col nuovo dividendo; in particolare cominceremo col vedere quante volte la prima cifra del divisore è contenuta nella prima cifra del nuovo dividendo. 2 : 1 dà 2 che collocheremo nella B, quale seconda cifra del quoziente.

5/ Moltiplichiamo la prima cifra del divisore (1) per l'ultima cifra del quoziente calcolato (2). Il risultato sarà 2 che sottrarremo dalla colonna D.

6/ Moltiplichiamo la seconda cifra del divisore 2 per l'ultima cifra calcolata del quoziente (2) e sottraiamo il risultato (4) dalla

posizione E, ottenendo il nuovo dividendo parziale 36.

7/ La prima cifra del nuovo dividendo (3) contiene 3 volte la prima cifra del divisore 1. Posizioniamo 3 sulla destra delle cifre di quoziente già calcolate (colonna C).

8/ Moltiplichiamo la prima cifra del divisore (l'1 di 12) per l'ultimo valore di quoziente trovato (3) e sottraiamo il risultato dalla E.

9/ Moltiplichiamo il 2 della seconda cifra del divisore sempre per l'ultima cifra del quoziente provvisorio (3) e sottraiamo il 6 risultante dalla F. Leggiamo quindi il risultato. Ecco schematicamente la successione dei passi descritti.

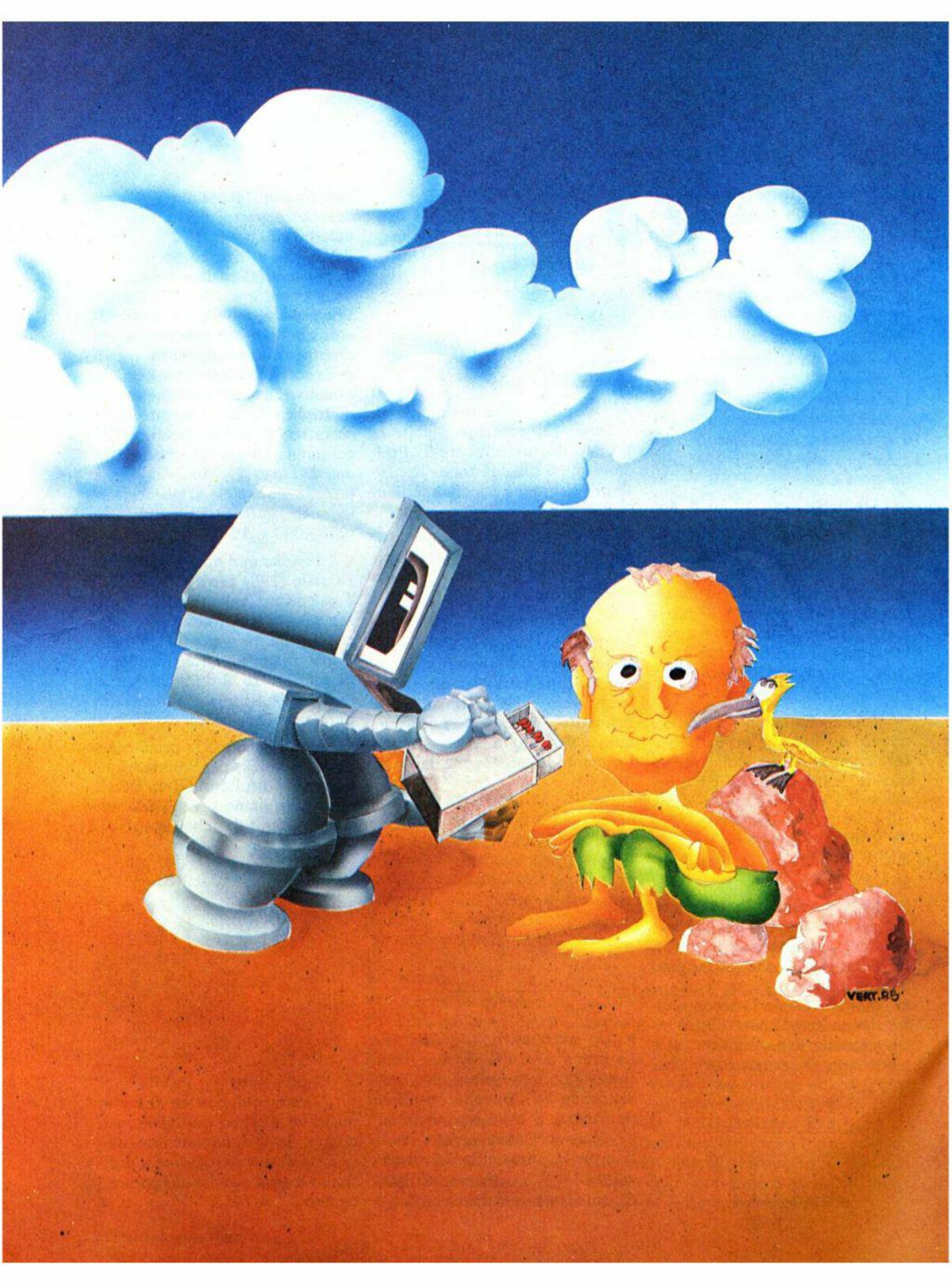
• *Un altro esempio: 638 : 22.* La prima cifra del divisore (2) è contenuta nella prima cifra del dividendo (6) 3 volte, per cui riportiamo 3 nella colonnina A.

Moltiplichiamo questo risultato per la prima cifra del divisore (2x3 = 6), sottraendo 6 dalla C; quindi per la seconda cifra (2) del divisore. Il risultato (6) andrebbe sottratto dalla cifra della colonnina D. Ma 6 è maggiore del 3 che vi ritroviamo; per cui dobbiamo (a) scalare d'una unità la penultima cifra di quoziente calcolata, quindi (b) riposizionare un 2 della prima cifra del divisore già moltiplicato per 3 nella corrispondente colonnina di calcolo C; successivamente si moltiplica la cifra di quoziente corretta (2) che ritroviamo sulla colonnina A per la seconda cifra del divisore (2). 2x2 dà 4 che posizioniamo sulla D per sottrarlo dal quoziente provvisorio (238 - 040 = 198).

Dividiamo il nuovo dividendo parziale (198) per il divisore (22) e calcoliamo quante volte la prima cifra di quest'ultimo è contenuta nella prima cifra (o nelle prime due cifre del dividendo, se maggiore). 18 : 2 dà 9 che segneremo sulla B, quale ultima cifra di quoziente calcolata.

Moltiplichiamo la prima cifra del divisore (2) per 9, quindi sottraiamo il risultato 18 da DE. Il quoziente, come si vede dalla seguente ricapitolazione, sarà 29.

Capito tutto?

Certamente, con un abaco in mano, vi sarebbe più facile esercitarvi. Se ne desiderate uno, non vi resta che profittare delle prossime vacanze e andare a comprarne uno . . . in Cina. ■

IL COMPUTER INTELLIGENTE

Ovvero come costruirsi un robottino in grado di imparare a giocare

Siete stesi sulla spiaggia e non avete niente da fare? Basta poltrire! Andate a rubare dagli ombrelloni vicini tutte le scatole di fiammiferi svedesi che potete, procuratevi forbici e colla e costruitevi IMP, il robot pensante al quale insegnate a giocare all'esapedone e che dopo un po' vi batterà con una regolarità esasperante.

Il campo di battaglia, costituito dagli incroci di una scacchiera, è considerato da sempre uno dei test più importanti delle capacità logiche di un essere umano.

Il celebre maestro di scacchi sovietico Botvinnik, ipotizzò che con l'approssimarsi dell'anno 2000 sarebbe stato un calcolatore elettronico il campione del mondo di questa specialità. A causa della sua affermazione, fu oggetto di una feroce critica da parte degli ambienti scacchistici internazionali.

A distanza di parecchi anni, siamo in grado di dirimere completamente l'argomento della disputa? Almeno 3 motivi ci spingono a ritenere sensate le argomentazioni di questo celebre maestro sovietico.

- Un computer non è soggetto ad errori di distrazione, come invece capita per quasi tutti gli esseri umani.
- Può valutare l'evoluzione delle varie fasi di gioco, ad una velocità assai superiore rispetto a quella ipotizzabile per qualsiasi cervello umano.
- Un calcolatore può apprendere dai propri errori, modificando il suo comportamento nel caso che in un successivo incontro si verificasse una situazione uguale o analoga a quella attualmente in corso.

I primi 2 punti non dovrebbero suscitare eccessivi dubbi nel lettore, mentre potrebbero esistere delle perplessità sulla veridicità della terza affermazione. La possibilità di modificare il proprio comportamento, a causa di una determinata esperienza effettuata, è sempre stata ritenuta una prerogativa prettamente umana.

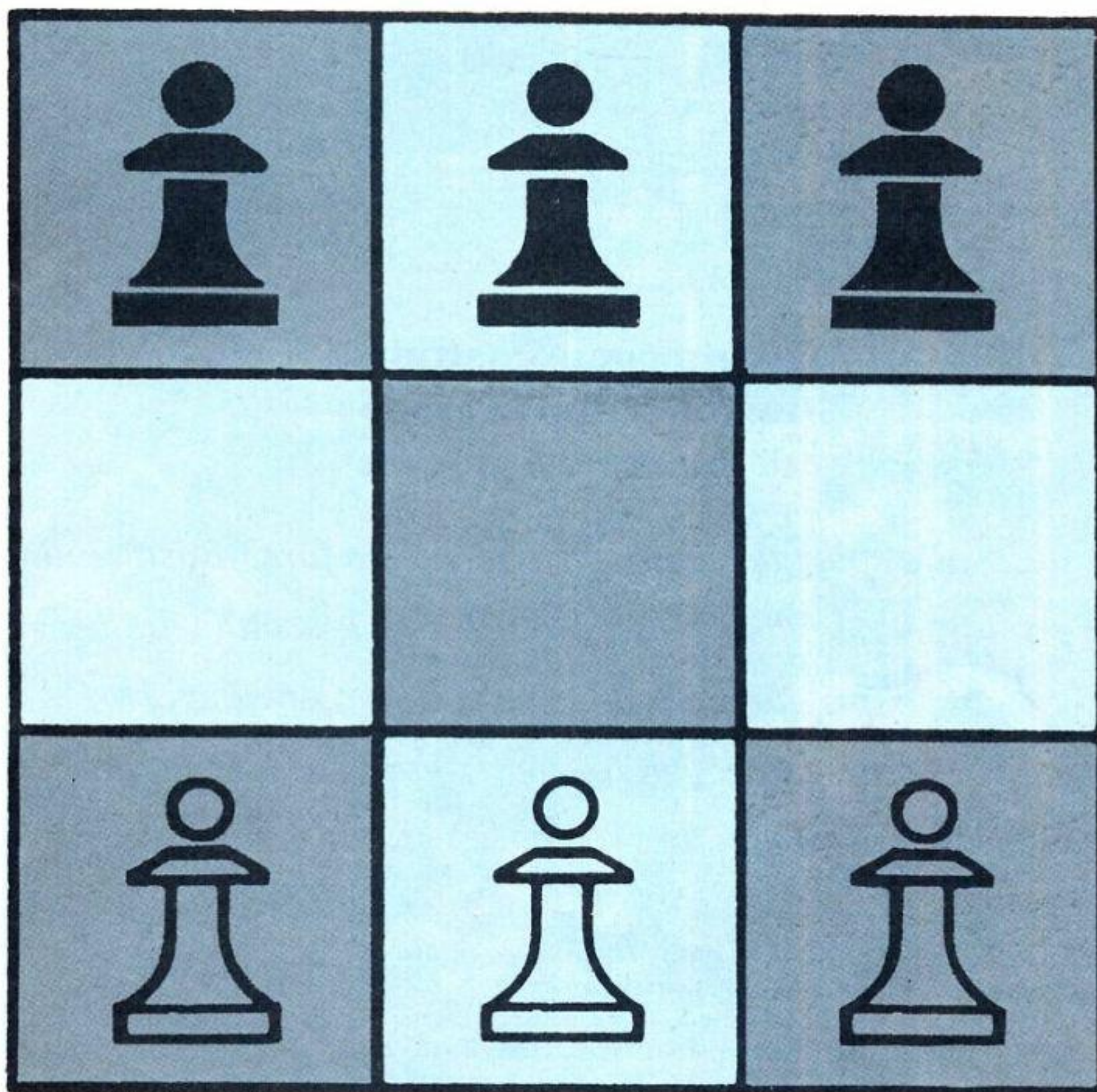
Se toccando un recipiente bollente, un bambino prova una sensazione di dolore, nel caso che si ripresentasse una situazione del genere eviterbbe il contatto con l'oggetto incandescente. Questo è un semplice esempio di come, soprattutto nella giovane età, lo sviluppo mentale di un'individuo è connesso agli stimoli che gli provengono dall'ambiente circostante.

Il terzo punto delle precedenti argomentazioni, inerenti al gioco degli scacchi, può presentare stratta analogie con l'analisi appena effettuata.

Se una macchina giocasse inizialmente in maniera assai scadente al gioco degli scacchi, ma avesse la possibilità di evolvere la propria tattica tramite le successive sconfitte riportate, il suo processo di miglioramento potrebbe non conoscere limiti.

Per meglio esemplificare i processi di auto apprendimento, affronteremo il tema dell'esapedone.

Abbiamo scelto questo genere di gioco, perchè riteniamo, che seppur molto semplice, sia indicato per la trattazione dei processi di intelligenza artificiale.

La competizione si svolge su una scacchiera delle dimensioni di 3 caselle per 3, con 3 pedoni situati su ciascun lato della superfice di gioco (osservate la figura 1).

Il gioco dell'esapedone.

Gli spostamenti possibili sono solo due:

* Il pedone può muoversi in avanti di una posizione.

• Un pedone può catturarne uno nemico, spostandosi in diagonale. La cattura di un pezzo provoca la sua immediata esclusione dal gioco in corso.

La competizione ha termine quando si verifica una delle 3 seguenti eventualità:

• Uno dei 2 contendenti è impossibilitato a muoversi senza contravvenire alle regole precedentemente citate.

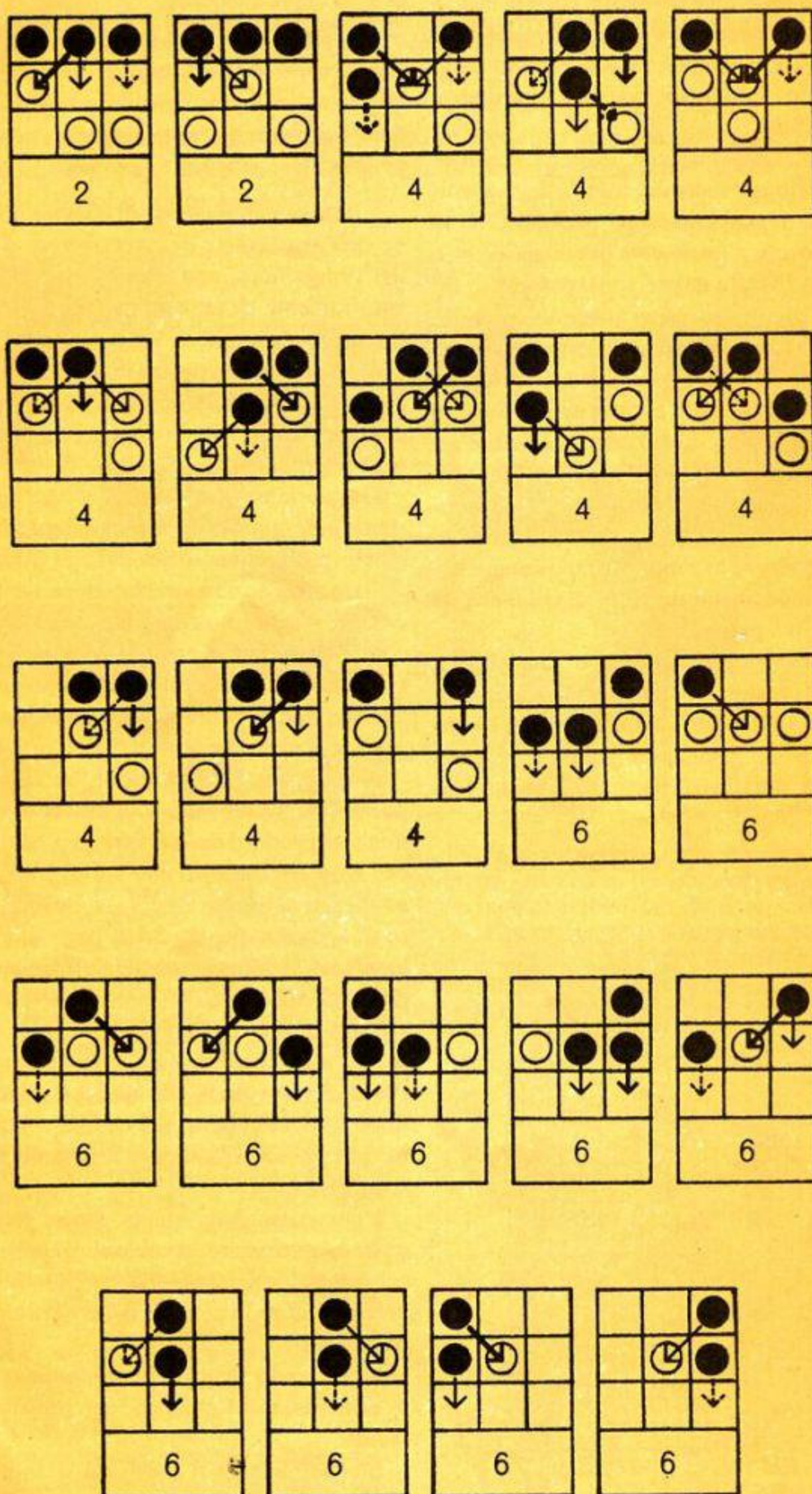
• Portando un proprio pedone sulla terza riga avversaria.

• Catturando tutti i pezzi nemici.

La durata del gioco non può essere naturalmente superiore alle 6 mosse complessive, sommando quelle effettuate dai due contendenti.

Ora vi illustreremo un processo di auto apprendimento, basato sul gioco dell'esapedone.

Prima di addentrarci nella spiegazione dell'argomento, è necessario munirsi del materiale di cui faremo uso nel seguito dell'articolo. Per costruire questa macchina pensante, non dovremo far ricorso obbligatoriamente ad un calcolatore, ma potremo avvalerci di semplici oggetti presenti in qualsiasi abitazione domestica.

Reperite 24 scatole di fiammiferi e alcune biglie colorate di dimensioni ridotte. Su ciascuna delle confezioni, deve essere incollato uno degli schemi riportati nella figura 2. Adesso ponete in ognuna delle scatole di fiammiferi, un numero di biglie pari alla quantità di freccette raffigurate sul diagramma incollato nella parte superiore di ciascuna confezione. Se ad esempio nello schema riportato su una scatola sono presenti 3 frecce, dovrò porre al suo interno 3 palline di colore differente. Vi raccomandiamo di abbina-

Contrassegni delle scatole di fiammiferi dell'IMP (i quattro diversi tipi di frecce rappresentano i quattro diversi colori).

re a ciascuna delle tre possibili frecce, una biglia di croma diverso.

Le operazioni preliminari sono esaurite, ora siamo in possesso di 24 scatole con i rispettivi diagrammi incollati nella parte superiore e contenenti un numero di palline pari a quello delle freccette.

La competizione comincerà tra noi e la macchina pensante (per rendere maggiormente familiare il rapporto con essa, supponiamo di denominarla IMP, acronimo di "intelligente macchina pensante"). Impostiamo la nostra prima mossa, dopo di che spetterà a IMP effettuare il secondo turno di gioco. Come avrete senza dubbio notato, alla base di ciascun diagramma, incollato sulle scatole di fiammiferi, è riportato un numero (2,4 oppure 6). Poichè IMP gioca sempre per seconda, queste cifre indicheranno tra quali alternative, o per meglio dire confezioni, è possibile scegliere in rapporto al numero della mossa di gioco attualmente in corso.

Ad esempio, nel secondo turno, IMP ha a disposizione 2 differenti scatole di fiammiferi. Dopo aver scelto casualmente una delle 2 confezioni, estraiamo da essa una delle biglie ed effettuiamo la mossa corrispondente alla freccetta abbinata alla pallina. La biglia estratta, deve essere posta al di sopra della scatola di fiammiferi da cui è stata tolta.
Adesso toccherà a noi effettuare la terza mossa. Dopo averla terminata, IMP effettuerà il suo quarto turno di gioco. In questo caso dovremo scegliere tra 11 possibili scatole di fiammiferi ed estrarre da una di esse una pallina colorata.
Dopo aver effettuato queste operazioni ed aver eseguito la mossa corrispondente al colore della biglia, poniamo la pallina al di sopra della scatola da cui è stata precedentemente estratta. Effettuiamo la nostra quinta mossa. Se non si è verificata una delle condizioni che pongono termine allo scontro tra i 2 contendenti, IMP effettuerà il sesto ed ultimo turno di gioco.

Le scatole a disposizione per la scelta dell'elemento finale sono 11, e sarà necessario ripetere una procedura del tutto

analoga a quella che ha caratterizzato le due precedenti mosse della nostra macchina pensante.

Adesso è terminata la partita e si possono naturalmente presentare due diverse eventualità:

- Noi siamo i vincitori
- Oppure ha vinto IMP.

Nel caso che vinca la macchina pensante, riponete tutte le biglie estratte, all'interno delle rispettive scatole di provenienza. Se invece voi risultaste vincitori, punite IMP confiscando la biglia che rappresenta la sua ultima mossa e non utilizzatela più nei successivi confronti. Al contrario le altre palline estratte dovranno essere reinserite all'interno delle confezioni da cui provengono.

Queste competizioni dovranno essere ripetute svariate volte e permetteranno al nostra IMP di eliminare tutte le mosse che si sono rivelate fatali ai fini del risultato finale del gioco. Se al termine di una sfida, una confezione di fiammiferi rimanesse priva di elementi, dovreste eliminarla dalle successive competizioni e togliere dal gioco la biglia dell'estrazione immediatamente precedente.

La macchina pensante, che era inizialmente incapace di giocare, escluderà, progressivamente, tutte le varianti che la hanno condotto ad una inevitabile sconfitta.

E' innegabile che IMP ha acquisito esperienza facendo tesoro degli errori passati, così come una persona evita di incorrere nuovamente in situazioni che si sono rivelate deleterie nel passato.

L'algoritmo che vi abbiamo proposto, dovrebbe consentirvi di costruire un programma di intelligenza artificiale, finalizzato al gioco dell' esapedone. Le migliori realizzazioni, dovrebbero consentire già dopo circa 20 partite, di avere un programma che gioca assai bene ad esapedone (vedere figura 3). L'acquisizione di esperienza, lo renderà praticamente imbattibile anche da parte di colui che ha creato il programma.

Attenzione!! Lanciamo una sfida a tutti i lettori di Commodore Computer Club, invitandoli ad elaborare ed a inviarci dei programmi, che prendendo spunto dall'algoritmo precedente illustrato, riescano ad elaborare un software di auto apprendimento.

Ora che si avvicinano le vacanze, perchè non provate ad utilizzare il tempo libero in montagna o sotto l'ombrellone, dilettandovi con questa simpatica competizione che vi abbiamo proposto?

Effettivamente la fase di comprensione dell'algoritmo e di successiva stesura del programma, non necessitano obbligatoriamente di un calcolatore. Alcune scatole di fiammiferi, un po' di palline colorate e svariate pagine di un blocco di appunti, vi permetteranno di passare alcune gradevoli ore in compagnia della nostra amica IMP.

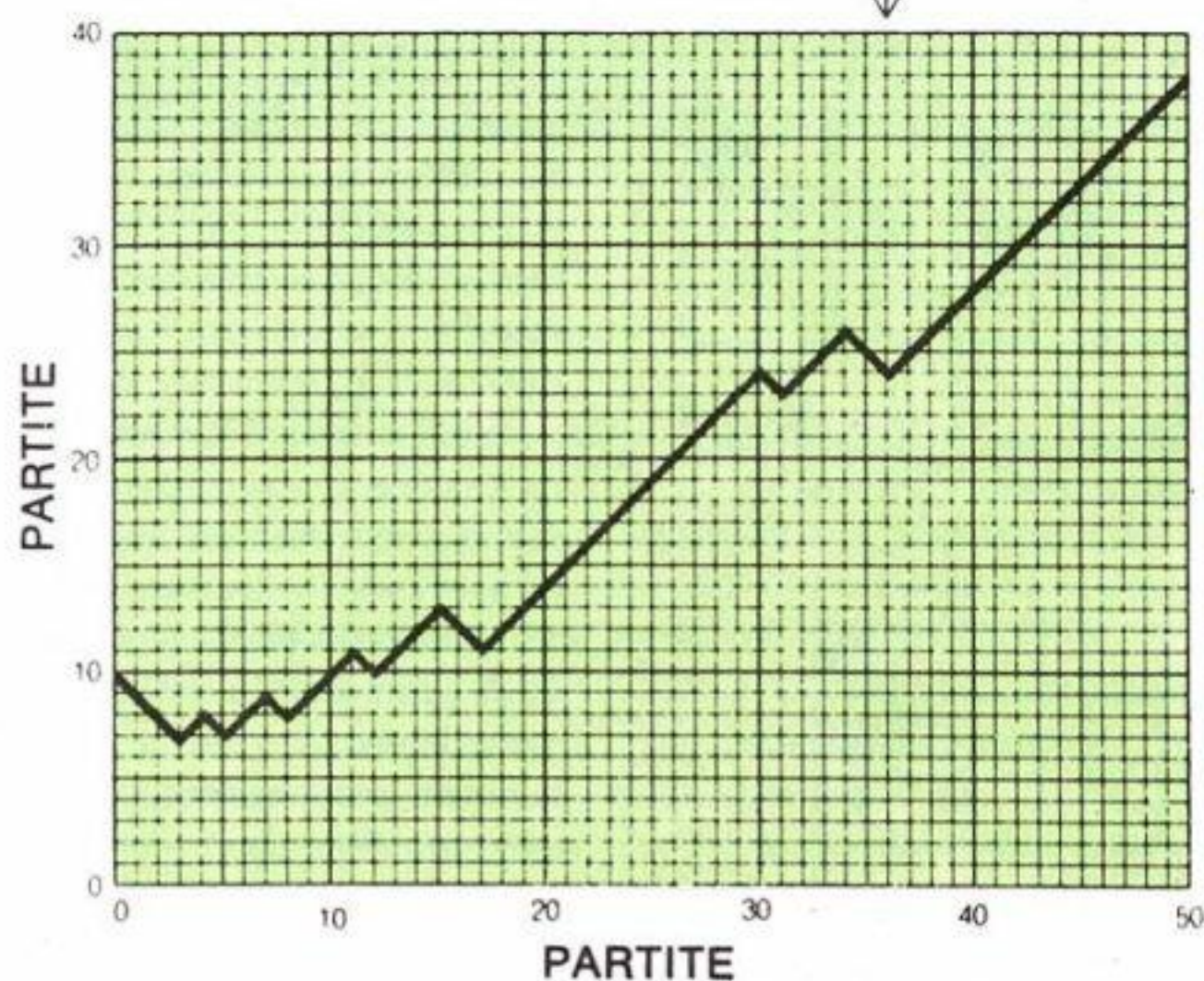
Desideriamo puntualizzare, che non siamo interessati ad un programma che giochi perfettamente sin dal suo primo utilizza, poichè non sarebbe concernente con l'argomento trattato in questo articolo. I contributi di coloro che si volessero cimentare su argomenti differenti dall'esapedone, risulteranno comunque graditi.

Le migliori elaborazioni verranno pubblicate sulla rivista e premiate con i migliori prodotti della nostra casa editrice. Non dimenticate che gli argomenti analizzati in questo articolo, sono oggetto di approfondito studio da parte di coloro che si interessano dell'evoluzione dell' informatica.

Tutti coloro che hanno sempre ritenuto che l'unico pregio di un calcolatore possa essere quello di eseguire grossi quantitativi di operazioni ad alta velocità, potrebbero risultare definitivamente sconfessati.

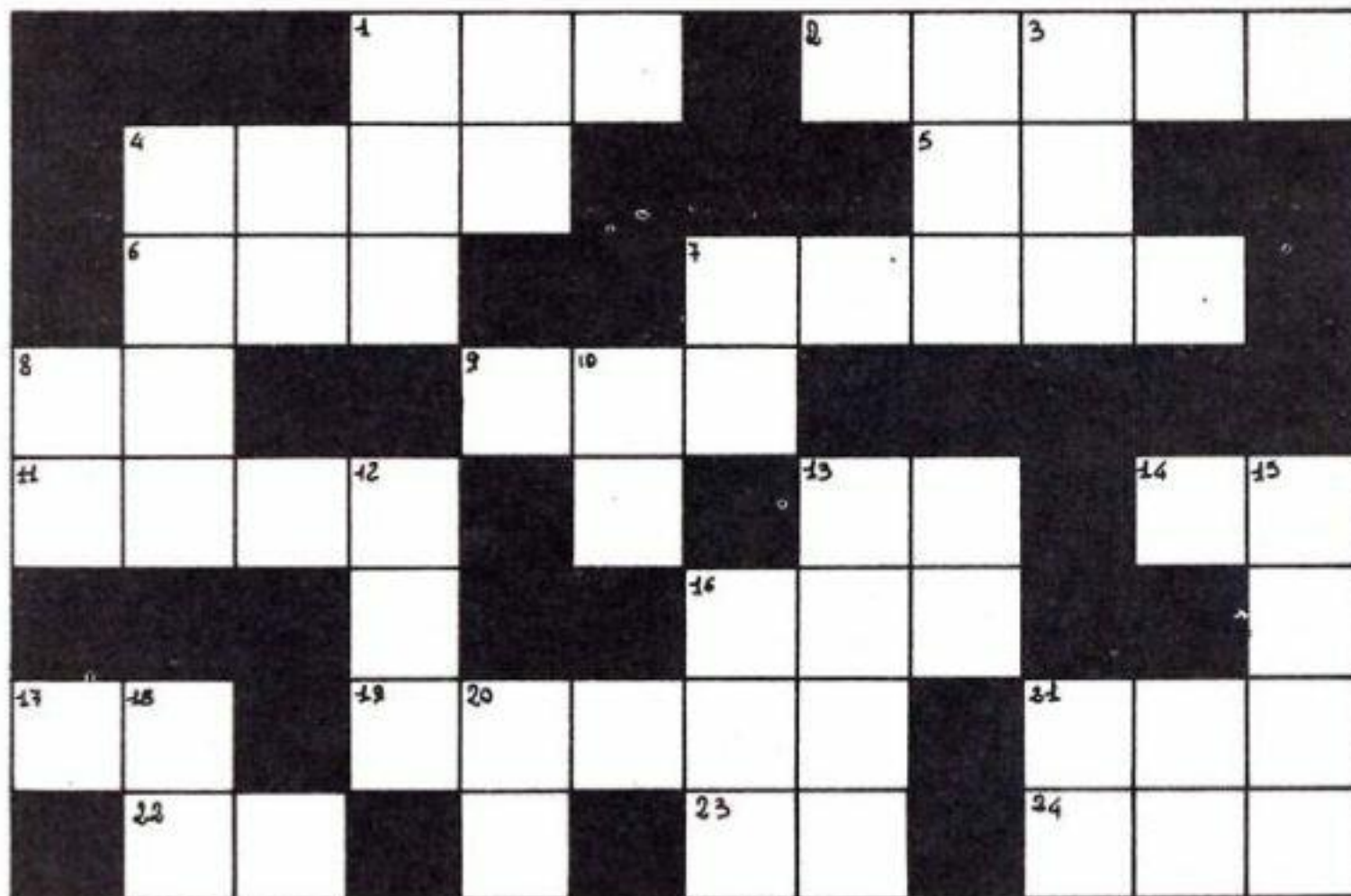
I computer della quinta generazione potranno acquisire esperienza dai propri errori? E' ipotizzabile una macchina che prenda autonomamente delle decisioni importanti?

Questi sono alcuni dei temi dominanti il panorama informatico dei prossimi anni.

Inevitabilmente il problema scientifico e quello etico dovranno trovare dei punti di riflessione comune, per evitare che l'uomo risulti impreparato alla sfida tecnologica dell'anno 2000.

Curva di apprendimento dell'IMP nelle prime cinquanta partite (i tratti inclinati in basso indicano sconfitte quelli in alto, vittorie.

Le parole incrociate di Commodore Computer Club

Mettete alla prova la vostra conoscenza del computer con questo semplice schema di parole incrociate. Siamo sicuri che la maggior parte di voi lo risolverà molto facilmente!

soluzioni a pag. 51

ORIZZONTALI

1- Misura il numero dei caratteri di una stringa. 2- Comando del C-16 con tre argomenti. 4- Restituisce il contenuto di una locazione di memoria. 5- Abbreviazione lecita con cui assegnare il valore ad una variabile numerica. 6- Prima parola del messaggio di errore che compare con: DIM A(10000). 7- Dopo che il programma termina. 8- Semplice messaggio di conferma. 9- Dopo IF nel caso sia diverso. 11- Genera un errore se privo di for. 13- Abbreviazione lecita per salto incondizionato Basic. 14- Un linguaggio oltre il Basic. 16- Vi è memorizzato il Sistema Operativo. 17- A patto che. 19- Ordine di visualizzazione. 20- Sigla di un noto circuito integrato del Vic. 21- Codice operativo. 22- Parente stretto del ciclo FOR... NEXT. 23- Le tre voci elettroniche del Commodore 64.

VERTICALI

1- Comando superfluo nei moderni linguaggi Basic. 3- Arrossisce quando registriamo. 4- Un suo uso improprio combina seri guai! 8- Può esserci prima di un salto. 10- Eseguire anche in altri casi. 12- Lo sono, della memoria, le locazioni 55 e 56. 13- Salta senza discussioni. 15- Riporta un programma in memoria. 16- Funzione indispensabile per programmare giochi alla roulette. 18- Abbreviazione prima di un ciclo. 20- Abbreviazione della piastra del comando iniziale. 20- Abbreviazione della piastra elettronica.

ESTATE: TEMPO DI QUIZ

Sia che ti trovi sotto l'ombrellone, sia che ti riposi in un bosco, sia che sei in casa a studiare per settembre (ahia!) metti alla prova la tua confidenza col computer.

Se possiedi un computer Commodore dovresti rispondere correttamente al primo gruppo di dieci domande ed al secondo gruppo che riguarda il tuo personal in particolare.

Più ne indovini, meglio è. Il consueto "punteggio" che viene abitualmente pubblicato alla fine di ogni quiz che si rispetti, nel caso presente non c'è.

Se proprio ci tieni a determinare il punteggio relativo alla tua preparazione in fatto di computer c'è un sistema più semplice e... segreto: mettiti davanti ad uno specchio e fatti l'esame di coscienza (doppio ahia!).

Avvertenza superflua: alle domande è necessario rispondere senza avere vicino il computer, ma semplicemente immaginando ciò che il computer risponderebbe digitando ciò che le domande richiedono.

Primo gruppo di 10 domande

Nota bene: (R) = premere il tasto Return.

1 - Provando a digitare:
PRINT VIVA LA PAPPA (R)
si ottiene in risposta:
a/ SYNTAX ERROR
b/ ILLEGAL QUANTITY ERROR
c/ 0
d/ VIVA LA PAPPA

2 - Lanciando il seguente programma:
10 GOSUB 10
si ottiene in risposta:
a/ Nessuna risposta perchè il programm a "gira" all'infinito.
b/ Il Crash del sistema.
c/ ILLEGAL QUANTITY ERROR
d/ OUT OF MEMORY ERROR

3 - Eseguendo l'istruzione:
PRINT CHR$(10*20-181)
si ottiene come effetto:
a/ Nessun effetto
b/ ILLEGAL QUANTITY ERROR
c/ Il cursore si sposta in alto a sinistra
d/ Cambia il colore del cursore

4 - Supponiamo che in una linea di un listato Basic figuri l'istruzione END. Ovviamente, quando il programma la "incontra", si ferma e compare la parola READY.

Se a questo punto digitiamo CONT (R) si ottiene:
a/ CAN'T CONTINUE ERROR
b/ ILLEGAL DIRECT ERROR
c/ Il programma continua l'esecuzione a partire dall'istruzione successiva ad END.
d/ Il programma riparte nuovamente come se fosse stato impartito il comando RUN

5 - In un listato Basic vi è la seguente riga:
100 INPUT A

Se (alla comparsa del punto di domanda tipico dell'INPUT) rispondiamo premendo un tasto alfabetico e il tasto Return, succede che:
a/ Il programma si ferma con un SYNTAX ERROR.
b/ Compare il messaggio REDO FROM START ed il programma si ferma.
c/ Compare il messaggio REDO FROM START e viene nuovamente posta la domanda.
d/ Alla variabile A viene associato il valore nullo ed il programma prosegue normalmente.

6 - Allo scopo di visualizzare un listato d cento righe numerate da 100 a 20C digitate:
LIST 100 - (100+100) (R)

L'effetto che ne risulta è il seguente:
a/ SYNTAX ERROR
b/ Viene visualizzata soltanto la riga 100
c/ Compare il listato e, subito dopo, un SYNTAX ERROR
d/ Viene effettuato il calcolo 100-(100+100)

7 - L'istruzione LET è poco usata dai Commodoriani perchè:
a/ Tale istruzione non esiste nel set di istruzioni Commodore.
b/ E' un'istruzione superflua.
c/ E' necessaria per definire funzioni matematiche molto rare.
d/ Viene usata solo per trasmissioni via modem.

8 - Nei computer Commodore vi sono due set di caratteri: Maiuscolo - minuscolo; Maiuscolo - Semigrafico visualizzabili solo uno alla volta. E' possibile visualizzarli contemporaneamente entrambi?
a/ No, mai.
b/ Sì, ma ricorrendo alla tecnica della ridefinizione dei caratteri.
c/ Sì, a patto di costruirli uno per uno in alta risoluzione.
d/ Sì, ma rinunciando ai caratteri in Reverse e ricorrendo alla programmazione dei caratteri.

9 - Ricorrendo all'istruzione POKE è possibile, per errore, tentare di trascrivere un dato in una zona di memoria ROM anzichè RAM. L'effetto che ne risulta è il seguente:
a/ Si rischia di bruciare il circuito integrato ROM interessato all'ɔperazione.
b/ Si ottiene il Crash del sistema.
c/ L'istruzione viene eseguita ma il dato non memorizzato.
d/ Compare il messaggio ILLEGAL QUANTITY ERROR.

10 - Il microprocessore dei computer Commodore "lavora" alla velocità di oltre un milione di operazioni al secondo. Col termine OPERAZIONE si intende:
a/ La singola istruzione del programma Basic.
b/ La singola istruzione di programma in Linguaggio Macchina.
c/ Ciclo macchina del microprocessore.
d/ Calcolo di operazioni binarie.

Secondo gruppo di domande

Commodore 64

1 - Per programmare correttamente un suono è necessario settare nell'ordine:
a/ Volume, A/D, S/R, frequenza nota alta, frequenza nota bassa, tipo di onda, numero voce.
b/ Numero voce, frequenza nota alta, frequenza nota bassa, volume, A/D, S/R, tipo di onda.
c/ Qualunque successione purchè il tipo di onda sia l'ultimo POKE dell'impostazione effettuata.
d/ Qualunque successione, purchè il primo POKE sia quello relativo al volume.

2 - Per programmare uno Sprite sono necessari 63 byte ed è possibile, quindi, memorizzarne uno per ciascun gruppo di 64 byte indicando al C-64 (mediante una POKE) il numero del blocco cui ci riferiamo.

Tenendo conto di quanto detto è possibile (senza ricorrere a "trucchi" particolari) memorizzare uno sprite:
a/ In una qualsiasi zona di memoria.
b/ Solo nell'area 49152 - 53248
c/ Al massimo fino a 16348
d/ Ovunque purchè ci si accontenti di sprite non multicolor.

3 - Il reset software del sistema si ottiene con:
a/ SYS 49142
b/ SYS 64738

c/ POKE 0,0
d/ NEW

4 - Un vostro amico ha acquistato un Commodore 64 modello Executive. Non vedete l'ora di scambiarvi programmi e vi recate a casa sua con le vostre cassette ma, al momento di mettervi al lavoro, vi accorgete che:
a/ I due computer sono totalmente incompatibili.
b/ Non è possibile caricare un programma da nastro perchè l'Executive non prevede il connettore del registratore.
c/ Alcuni programmi sembrano non girare correttamente.
d/ Lo scambio di programmi avviene regolarmente.

5 - Un vostro amico possiede un computer Sinclair Specrum e vi chiede di far girare sul vostro Commodore 64 i suoi programmi su nastro. Voi rispondete che:
a/ L'operazione è del tutto impossibile a causa della completa incompatibilità dei due computer.
b/ L'operazione è possibile solo per programmi che richiedono una memoria di 16K RAM.
c/ L'operazione è possibile perchè esiste, per Commodore 64, il programma di simulazione Spectrum.
d/ L'operazione è possibile purchè i programmi siano scritti in Basic e non in Linguaggio Macchina.

Commodore 16

1 - La zona di memoria video del C-16 occupa 1000 celle a partire dall'indirizzo:
a/ 1024
b/ 2048
c/ 3072
d/ 65536

2 - Un programma scritto in Basic utilizzando un Commodore 64 ha molte probabilità di girare sul C-16 a patto che:
a/ Non contenga istruzioni POKE, SYS, PEEK
b/ Sia lungo almeno 20 linee Basic.
c/ Venga compilato.
d/ Non contenga istruzioni PRINT e INPUT.

3 - Un listato scritto con un Vic 20 oppure con un Commodore 64 può essere trasferito sul C-16 tramite:
a/ Registratore.
b/ Drive per minifloppy.
c/ Connessioni elettriche tra i due computer.
d/ Interfaccia RS-232 C

4 - Prima di utilizzare le istruzioni grafiche in alta risoluzione è necessario:
a/ Cancellare sia lo schermo "normale" sia quello in alta risoluzione.
b/ Utilizzare l'istruzione GRAPHIC.
c/ Colorare di nero il fondo ed il bordo del video.
d/ Ricorrere alle istruzioni GOSUB e COLOR.

5 - Il tasto HELP serve per:
a/ Rintracciare gli errori di programmazione.
b/ Per uscire dalle situazioni di pericolo tipiche dei videogiochi.

c/ Per seguire i corsi di auto-apprendimento al calcolatore.
d/ Ripristinare le condizioni iniziali in caso di Crash del sistema.

Vic 20

1 - Per riprodurre gli sprite col Vic 20 è necessario:
a/ Aggiungere una scheda di espansione di memoria.
b/ Aggiungere una particolare scheda grafica prodotta dalla Commodore.
c/ Non è possibile generare sprite col Vic 20.
d/ Utilizzare un monitor invece di un televisore.

2 - Per cambiare colore al bordo e al fondo dello schermo è necessario utilizzare le locazioni:
a/ 36879 per il bordo e 36880 per il fondo.
b/ Il nibble alto di 36879 per il fondo e quello basso per il bordo.
c/ Il contrario della risposta b.
d/ Il contrario della risposta a.

3 - Il reset software del sistema si ottiene con:
a/ Tasto Run/stop e Restore.
b/ SYS 64802
c/ SYS 64738
d/ NEW

4 - Col Vic 20 senza espansioni non è assolutamente possibile:
a/ Utilizzare il drive per minifloppy.
b/ Utilizzare il joystick
c/ Lavorare in linguaggio macchina
d/ Far girare programmi più lunghi di 3K.

5 - Prima che uscisse di produzione, il prezzo ufficiale, I.V.A. esclusa, di listino del Vic 20 (versione base) era di lire:
a/ 225000
b/ 199000
c/ 169000
d/ 99 dollari perchè nel listino ufficiale il prezzo è sempre espresso in dollari.

Le risposte sono a pag. 80

OLD

Come recuperare un programma Basic perso per un accidentale New o a causa di un 'crash' del sistema.

Vediamo come un programma BASIC è memorizzato nella RAM del Cervellone oh, pardon, del Computer. Il primo byte è posto a zero, valore che corrisponde alla fine linea BASIC; i successivi 2 byte servono da puntatori all'indirizzo della linea seguente (per linea mi riferisco sempre a quella BASIC), gli altri 2 byte successivi corrispondono al numero di linea, con la classica legge del byte basso, byte alto. Seguono poi le istruzioni Basic (ovviamente 'Tokenizzate'), con alla fine un byte a zero, che come già detto indica la fine della linea.

Di tutto questo interessano soprattutto il secondo e terzo byte (vedi Fig. 1). Infatti quando diamo il comando NEW, il calcolatore azzera queste due locazioni di memoria per il semplice motivo che tre zeri consecutivi — per il Calcolatore — stanno ad indicare la fine del programma.

Vengono inoltre reinizializzati i puntatori di pagina zero che contengono gli indirizzi di inizio e fine programma Basic, di inizio zona variabili e di inizio stringhe (dal 45 al 52 vedi manuale CBM 64).
A questo punto credo sia chiaro che per ritrovare il programma scomparso (Ellery Queen non c'entra) basta riuscire a riposizionare il secondo e terzo byte del programma e a riportare ai valori precedenti i puntatori di pagina zero interessati.

È proprio quello che "OLD" fa.
Viene prima ricercato lo zero indicante la fine della prima linea (escludendo ovviamente i primi tre), dopodichè si scrive l'indirizzo successivo nei 2 byte che erano stati azzerati dal NEW (byte basso, byte alto). Fatto questo, viene la parte più lunga del lavoro (ma a noi non interessa, perché il lavoro lo fa 'Cervellone'): bisogna infatti ricercare la fine del programma 'saltando' di indirizzo in indirizzo fino a che non troviamo tre zeri consecutivi. È questo l'indirizzo che dobbiamo immettere nei puntatori Basic sopra detti ed avremmo finito.

Può comunque aiutare per comprendere quanto detto far riferimento al listato Assembler. In esso vengono usati, come puntatori temporanei, 4 byte di pagina zero che, grazie a 'Mamma Commodore', non vengono usati dal Sistema.

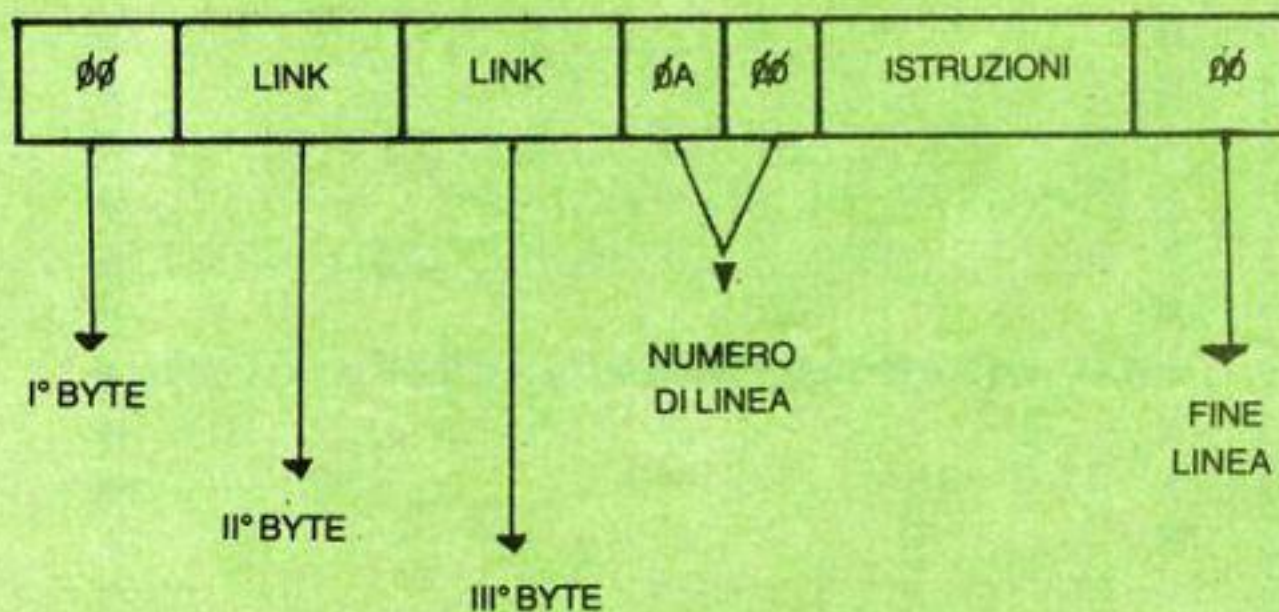



FIG. 1

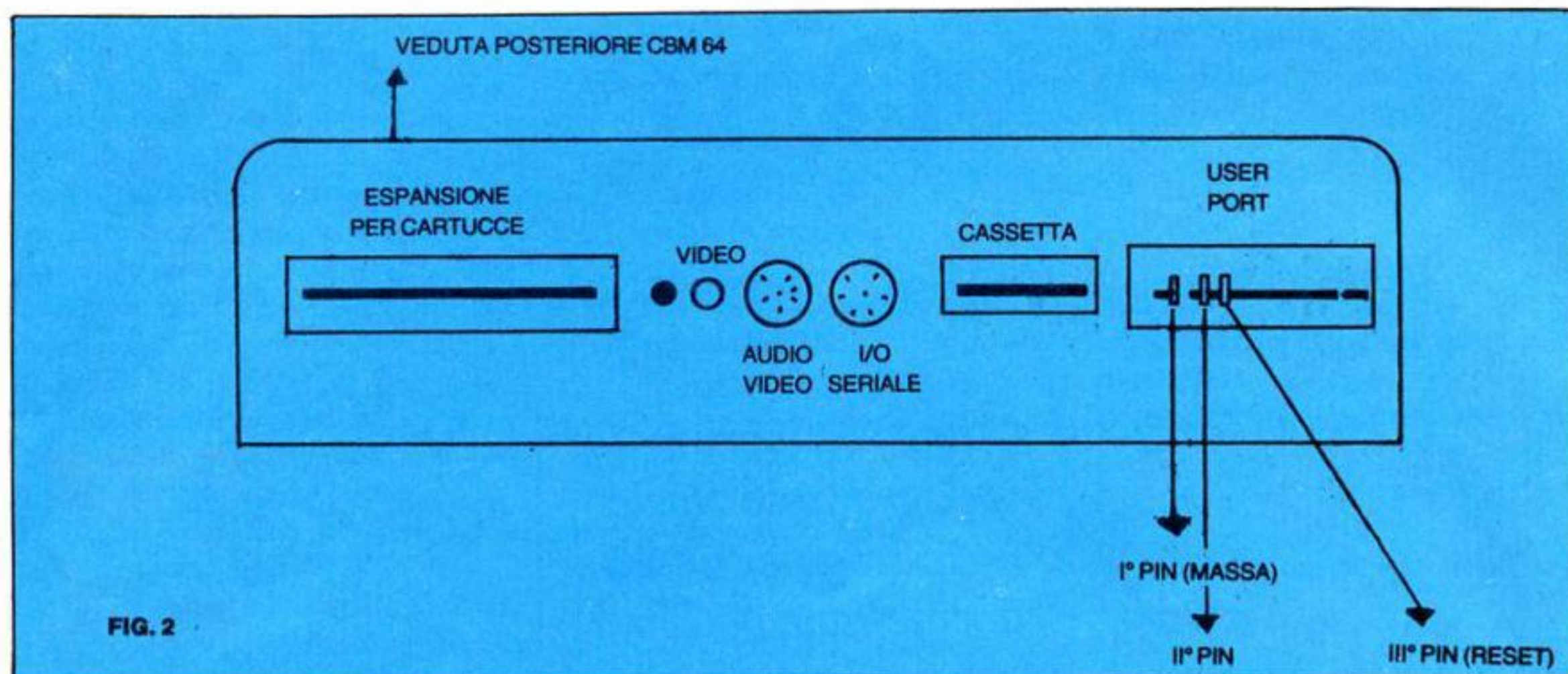



FIG. 2

Come si usa

Per l'utilizzo del Programma OLD basta, dopo aver digitato il listato BASIC (ed averlo registrato, non si sa mai!), dare il RUN ed eseguire le istruzioni per lo stesso programma. Infatti oltre a porre in memoria la routine LM di OLD, viene anche scritta una routine che salva su nastro o su disco OLD stesso.

A questo punto non resta che, quando se ne ha il bisogno (in caso di NEW accidentali), caricare in memoria OLD nel seguente modo:

load "old", 8, 1 se da disco
load "old", 1, 1 se da nastro
e, alla fine del caricamento, digitare 'SYS 53110' seguito dal RETURN.

Da notare che se viene usato il registratore come memoria di massa si può omettere il '1, 1' nel caricamento in quanto il programma viene salvato in maniera tale che non venga rilocato. Bisogna stare attenti a non commettere errori durante la fase di digitazione dei comandi per caricare e far eseguire il programma OLD, perché se ottenessimo un malaugurato SYNTAX ERROR dovremmo dire ADDIO al nostro listato Basic. Questo perché un tale errore sconvolge la RAM dove risiede il programma Basic rendendo impossibile un suo recupero! All'inizio dell'articolo ho detto che era possibile far 'risorgere' il programma Basic anche in caso di 'crash' del sistema (causato ad esempio da una sys errata), e mantengo la promessa.

In caso di 'crash' bisogna fare in modo di spegnere la macchina e riaccenderla senza toglierle l'alimentazione.

Come si può farlo in una macchina che, come il CBM 64, non possiede un tasto di RESET? Occorre portare a massa il terzo PIN della USER PORT, che guarda caso è il PIN di RESET.

In pratica basta far contatto con del materiale induttivo tra il 1 e il 3 PIN della USER PORT. Tanto per intenderci si immagini di vedere il CBM 64 da dietro e si faccia riferimento alla Figura 2.

```
1000 ;**********************************
1010 ;*                                *
1020 ;*            O L D               *
1030 ;*                                *
1040 ;* PERMETTE DI RECUPERARE UN      *
1050 ;* PROGRAMMA BASIC DOPO UN        *
1060 ;* ACCIDENTALE COMANDO DI 'NEW'   *
1070 ;*                                *
1080 ;**********************************
1090 ;*                                *
1100 ;* DI                             *
1110 ;*                                *
1120 ;* BEANI GIOVANNI                 *
1130 ;*
```

```
1200
1210
1220
1230
1240          *=53110          ;ESA $CF76
1250          ENBAS=$2D        ;PUNTATORE FINE BASIC
1260          RAM=$0800        ;INIZIO RAM BASIC
1270          PRINT=$AB1E      ;ROUTINE STAMPA
1280          LDY #5           ;RICERCA PRIMO ZERO
1290 LOOP1    LDA RAM,Y
1300          BEQ ZERO
1310          INY
1320          BNE LOOP1
1330          RTS
1340 ZERO     INY              ;POSIZIONA I PRIMI DUE
1350          STY RAM+1        ;BYTE DEL PROGRAMMA
1360          LDA #8           ;ALL'INDIRIZZO SUCCESSIVO
1370          STA RAM+2        ;ALLO ZERO TROVATO
1380
1390          LDY #8           ;RICERCA FINE PROGRAMMA
1400          LDX #1
1410          STX $FD
1420          STY $FE
1430 LOOP2    LDX $FD
1440          LDY $FE
1450          STX $FB
1460          STY $FC
1470          LDY #0
1480          LDA ($FB),Y
1490          STA $FD
1500          INY
1510          LDA ($FB),Y
1520          STA $FE
1530          BNE LOOP2
1540          LDA $FD
1550          BNE LOOP2
1560          LDA $FB          ;TROVATA FINE PROGRAMMA
1570          CLC
1580          ADC #2
1590          BCC EXIT
1600          INC $FC
1610 EXIT     LDY $FC
1620          STY ENBAS+1      ;POSIZIONA PUNTATORI
1630          STY ENBAS+3
1640          STY ENBAS+5      ;INIZIO VARIABILI
1650          STA ENBAS        ;INIZIO STRINGHE
1660          STA ENBAS+2      ;FINE BASIC
1670          STA ENBAS+4
1680
1690          LDA #<MSG        ;STAMPA MESSAGGIO
```

Sulla griglia elettronica, per salvarsi non bisogna incrociare le altre moto. Succede nei videogames. Ben diverso (e pauroso) sarebbe trovarsi all'interno di una delle moto. E' quello che succede a Flynn, il protagonista di Tron. Che però alla fine si salva, ai danni degli altri due contendenti. Il lieto fine è obbligatorio.

```
1700          LDY #>MSG
1710          JMP PRINT      ;E ESCE
1720 MSG      .BYTE ???,???,???,???,???
1730          .END

READY.
```

```
  1 REM  *************************
  2 REM  *          O L D        *
  3 REM  * DI                    *
  4 REM  *     BEANI GIOVANNI    *
  5 REM  *                       *
  6 REM  * TEL.0584/20670        *
  7 REM  *     V150884/CBM 64    *
  8 REM  *************************
  9 :
 10 ME=679 : I=0 : J=0
 20 PRINT CHR$(147);"ATTENDI ..."
 30 READ  A : IF  A<0  OR  A>255  T
    HEN  ON   SGN(A)+2 GOTO  50,,70
 40 POKE  ME+J,A : I=I+A : J=J+1 :
    GOTO  30
 50 IF  I+A<>0  THEN  PRINT "ERRORE
     DATA SAVE" : STOP
 60 ME=53110 : I=0 : J=0 : GOTO  30
 70 IF  I<>A  THEN  PRINT "ERRORE D
    ATA OLD" : STOP
 80 PRINT CHR$(147);"SALVI SU <N>AS
    TRO O SU <D>ISCO "; : INPUT  DE
    $
 81 IF  DE$="N"  THEN  POKE  683,1:
    GOTO 85
 82 IF  DE$<>"D"  THEN  80
 85 SYS 679 : REM  (C) BEANI GIOVAN
    NI
 87 PRINT : PRINT "FATTO !"
 88 END
500 :
510 REM  DATI ROUTINE 'SAVE'
520 :
679 DATA  169,003,168,162,008,032,1
    86
686 DATA  255,169,003,162,201,160,0
    02
693 DATA  032,189,255,162,118,160,2
    07
700 DATA  134,251,132,252,169,251,1
    62
707 DATA  000,160,208,076,216,255,0
    79
714 DATA  076,068,-5292
1000 :
1010 REM  DATI PROGRAMMA 'OLD'
1020 :
53110 DATA160,005,185,000,008,240,0
      04
53117 DATA200,208,248,096,200,140,0
      01
53124 DATA008,169,008,141,002,008,1
      60
53131 DATA008,162,001,134,253,132,2
      54
53138 DATA166,253,164,254,134,251,1
      32
53145 DATA252,160,000,177,251,133,2
      53
53152 DATA200,177,251,133,254,208,2
      35
53159 DATA165,253,208,231,165,251,0
      24
53166 DATA105,002,144,002,230,252,1
      64
53173 DATA252,132,046,132,048,132,0
      50
53180 DATA133,045,133,047,133,049,1
      69
53187 DATA201,160,207,076,030,171,1
      47
53194 DATA017,032,083,084,065,073,0
      32
53201 DATA065,084,084,069,078,084,0
      79
53208 DATA032,076,065,032,080,082,0
      79
53215 DATA083,083,073,077,065,032,0
      86
53222 DATA079,076,084,065,032,033,1
      41
53229 DATA141,032,071,046,066,069,0
      65
53236 DATA078,073,032,052,047,057,0
      47
53243 DATA056,052,141,15371
```

QUALSIASI COMPUTER

Come Realizzare un Menu'

Tutti gli ingredienti, i trucchi ed i suggerimenti per inserire nei vostri programmi la possibilità di scelta tra numerose opzioni.

La prima cosa che si trova sul tavolo, di un ristorante, è il menù: un libretto sul quale sono segnate tutte le 'specialità' della casa ed in base al quale si farà la propria scelta 'mangereccia'.

Nel linguaggio dell'informatica, il menù ha lo stesso significato (con un po' di elasticità mentale...). E' una lista di "compiti", che il programma è in gra do di assolvere, tra i quali possiamo scegliere con la semplice pressione di un tasto.

Scopo di questo articolo è appunto quello di spiegare come si può 'creare' un menù, passo dopo passo. Come esempio supponiamo di voler fare un programma che si occupi, dato un lato, di calcolare l'area di un quadrato oppure di un triangolo equilatero. In effetti, il compito del programma può sembrare banale, ma non vedo perchè complicarsi la vita (e la digitazione) con programmi più complessi, di difficile comprensione e di... identiche finalità!.

Le formule usate, saranno le seguenti:
Area quadrato: lato X lato
Area triang. eq.: lato X lato X sqr(3)/4

In figura 1, è illustrato il diagramma a blocchi che mostra lo schema principale del programma.

Vediamo ora come realizzare, blocco per blocco, le singole parti del diagramma.

Visualizza il menù

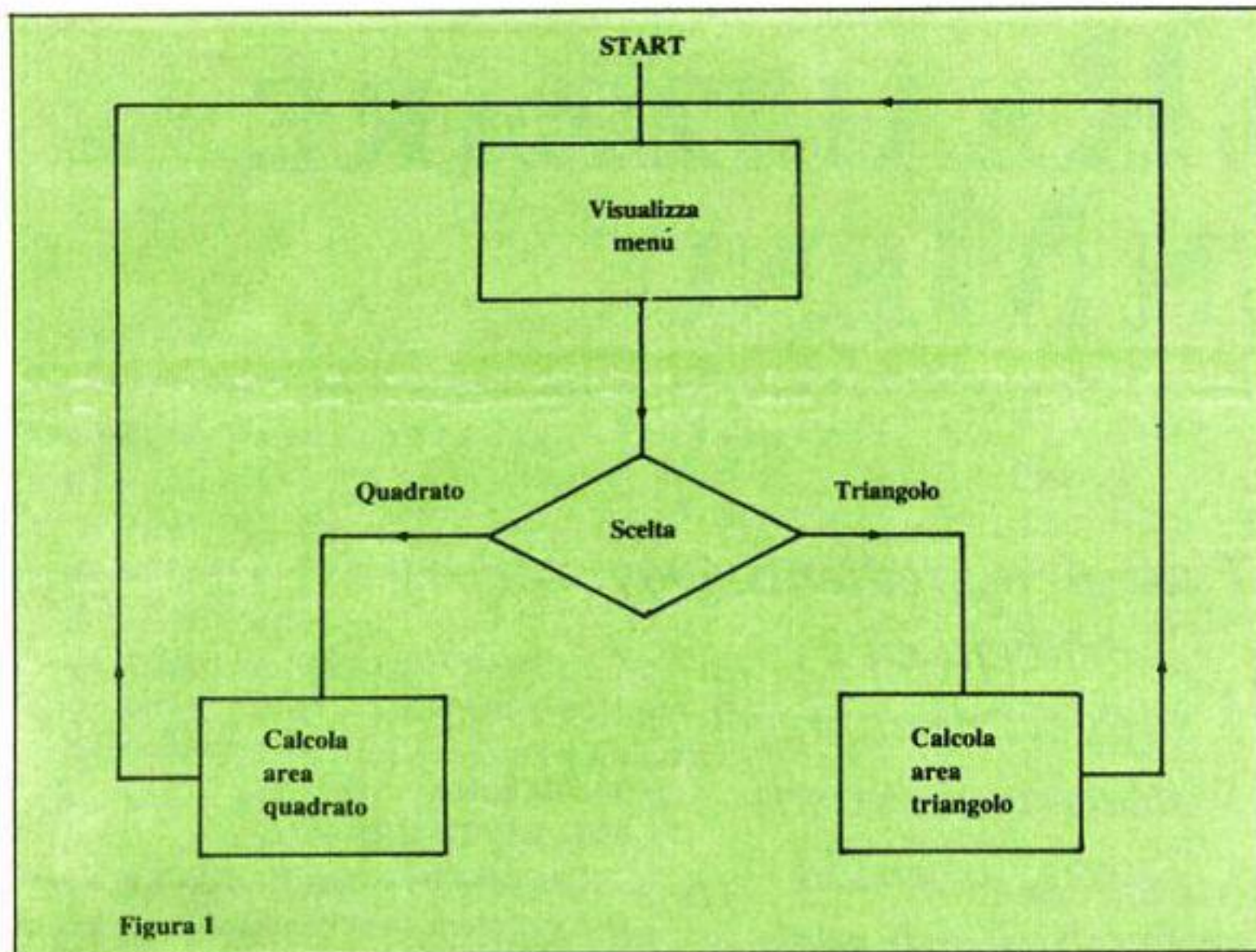
Come prima istruzione, digitate la seguente linea:
100 PRINT CHR$(147).

Ora provate a dare il 'RUN': lo schermo si azzera (=si cancella) e, in alto a sinistra, compare la scritta 'READY', a conferma che il programma è stato eseguito. L'istruzione CHR$(147), preceduta da PRINT, fa comparire sullo schermo quel carattere ASCII corrispondente al valore X assegnato. Nel nostro caso, il valore 147 corrisponde all'azzeramento dello schermo, cioè, in pratica, è l'equivalente di ciò che si ottiene premendo contemporaneamente i tasti (SHIFT) e (CLR - HOME).

Se sostituite il valore 19 al posto di "X" in CHR$(X), ottenete l'equivalente del tasto (CLR - HOME): il cursore si sposta in alto a sinistra, senza azzerare lo schermo.

Continuiamo il nostro programma, digitando ora le linee:
110 PRINT"QUADRATO 1"
120 PRINT"TRIAN.EQ. 2"
dove i numeri 1 e 2 saranno i tasti da premere per ottenere il calcolo relativo alla figura geometrica desiderata).

Per vedere cosa succede, date tranquillamente il 'RUN': in alto a sinistra compariranno le scritte "QUADRATO 1" e, sotto, 'TRIAN.EQ. 2'. Questo sarà il nostro menù. In effetti, la visualizzazione del menù deve anche soddisfare certe esigenze di carattere estetico (molto soggettive!), per cui modificate le tre linee scritte finora nel seguente modo:

Figura 1

```
100 PRINT CHR$(147) CHR$(17)
    CHR$(17) CHR$(17)
110 PRINT SPC(12) "QUADRATO"
    "CHR$(18)" 1 "CHR$(17)
120 PRINT SPC(12) "TRIAN.EQ."
    "HR$(18)" 2 "CHR$(17)
```

Da quanto potete vedere, si sono introdotti altri due valori X in CHR$(X). Il valore 17 è l'equivalente del tasto (CRSR - giù): in questo modo, nella linea 100, oltre ad azzerare lo schermo, si dice anche al computer di saltare tre righe.Il secondo valore, 18, fa sì che tutte le scritte e gli spazi tra virgolette che lo seguono, saranno stampate nel modo "REVERSE" (in negativo). Nel nostro caso, saranno stampati in reverse i numeri 1 e 2, avvolti tra due due spazi anch'essi in reverse. Dal punto di vista estetico, questo tipo di scrittura è molto usato e, fra l'altro, mette bene in evidenza i tasti da premere per ottenere l'opzione desiderata.

L'ultimo comando da analizzare in questa prima parte, è SPC(12). Se date il RUN, vi accorgerete da soli dell'effetto di questa istruzione! Ora le scritte non sono più visualizzate al margine sinistro dello schermo, ma sono spostate da esso di 12 caratteri (per fare altre prove potete cambiare questo valore con altri a vostro piacere).

E' Ovvio che, come detto in precedenza, la gestione della schermata di 'menù' è soggettiva, perciò, con l'aiuto dei pochi comandi visti finora, potete sbizzarrirvi a stampare le scritte come volete! L'unica cosa che vorrei sottolineare, è che i comandi CHR$(X) e SPC(Y) devono sempre essere preceduti da PRINT, altrimenti si 'cade' in un SYNTAX ERROR.

Scelta

Dopo aver visualizzato il menù, dobbiamo, far capire la nostra scelta al computer: cioè dobbiamo dialogare, tramite qualche istruzione, col computer stesso.

Incominciamo con la linea: 200 INPUT R dove R sarà la variabile che conterrà la nostra risposta: se vorremo il quadrato, premeremo il tasto 1 (e, subito dopo, RETURN) e la variabile R assumerà di conseguenza il valore 1, mentre se vorremo il triangolo, premeremo il tasto 2, e la variabile R assumerà il valore 2.

Se ora fate girare il programma, sotto il menù comparirà un punto di domanda e di fianco il solito cursore lampeggiante: questo è l'effetto dell'istruzione INPUT.

Di seguito, quando il programma incontra una INPUT, è di fermarsi, stampare il "?", ed aspettare una risposta, che deve sempre essere seguita dalla pressione del tasto +RETURN.

Come prova, digitate 1 (o un altro numero) e premete il tasto RETURN (operazione che d'ora in poi chiameremo semplicemente: (RET): sullo schermo comparirà il consueto READY, ad indicare l'avvenuta esecuzione del programma. Una volta scelta l'opzione, il computer deve essere in grado di identificarla e di comportarsi di conseguenza. Questo può essere fatto con istruzioni di tipo "logico", che permetteranno al nostro "cervellino elettronico" di agire in un modo oppure in un altro (ricordatevi sempre, comunque, che la logica è nostra, non del computer: siamo noi che stabiliamo quello che deve fare!).

L'istruzione atta a tale scopo, è IF..THEN: in pratica, dice che SE (IF) è verificata una certa condizione, ALLORA (THEN) fai una certa cosa. Se la condizione non è verificata, il programma continua normalmente, elaborando, cioè, la linea Basic successiva, senza preoccuparsi dell'IF..THEN.

Nel nostro caso, dobbiamo scegliere tra il quadrato (R=1) oppure il triangolo (R=2): possiamo farlo nel seguente modo:

```
210 IF R=1 THEN GOTO 500
220 IF R=2 THEN GOTO 600
:
:
500 PRINT "QUADRATO": END
600 PRINT "TRIANGOL": END
```

La linea 210 dice: se R=1, allora vai alla linea 500, dove una routine si occuperà di eseguire le operazioni sul quadrato. La linea 220, è analoga.

In questo modo, abbiamo visualizzato il menù ed operato la nostra scelta. Ma è ancora troppo poco: un programmatore deve prevedere anche i possibili errori che si possono compiere nel dialogare col computer.

L'errore più banale, a questo punto, è quello di digitare un numero diverso da 1 o 2: in questo caso, non verificandosi nessuna delle due condizioni (R=1 oppure R=2), il programma fallisce miseramente.

Quindi bisogna prevedere che se R è diverso da 1 o da 2, occorre tornare indietro e ripetere la risposta: si può ottenere ciò con la linea

```
230 IF R < > 1 AND R < > 2 THEN 200
```

Se R è < > 1 (diverso da 1) e contemporaneamente (AND) < > 2, allora ritorna alla linea 200, fino a che la risposta non è "accettabile" (1 oppure 2). Provate ora a dare il solito RUN: rispondendo con 1 otterrete la stampa della parola QUADRATO; premendo 2, otterrete TRIANGOL.

Se poi premete, ad esempio, 3, otterrete di nuovo il ?, cioè il computer sta aspettando ancora la risposta esatta. Dopo ogni prova positiva il programma si "ferma" (READY.) e dovete di nuovo dare il RUN.

Provate ora a premere, invece di un numero, una lettera qualsiasi: non vedrete comparire il solito READY ma la scritta REDO FROM START e sotto, di nuovo il punto di domanda (?). Finchè digiterete lettere, otterete sempre la stessa scritta. In effetti, il programma non si 'blocca' (non segnala errori), ma riceve una risposta che non si aspettava (carattere alfabetico invece che numerico) e risponde, quindi, con un 'ripetere dall'inizio'.

Di seguito se siete stufi del messaggio REDO FROM START, digitate pure un numero e tutto proseguirà normalmente.

Per capire cosa è successo, occorre spendere due parole sulle "variabili", ossia sui dati che introduciamo e che ci consentono il 'dialogo'.

Queste sono essenzialmente di due tipi: variabili numeriche e variabili alfanumeriche. Le variabili numeriche come del resto dice il nome, sono variabili di tipo numerico, ammettono cioè solo numeri.

Si possono indicare con lettere tipo R, S, AA, A1, etc, escludendo le var. 'vietate' (vedere art. 'Sintax Error' sul num. 17). Le variabili alfanumeriche, ammettono, oltre ai numeri, anche le lettere e i caratteri grafici, trattando tutto come caratteri (anche i Numeri!): si indicano con lettere seguite dal simbolo del dollaro ($). Diamo alcuni esempi:

Var. Numeriche	Var. Alfanumeriche
A = 10	A$= "PIPPO"
AA=1.8	AA$= "PLUT2O34"
A2=.01	A2$= "12348"

(un discorso più approfondito, sarà fatto in seguito)

Tornando al nostro programma, nella linea 200 avevamo scritto INPUT R, dove R è una variabile numerica. E' ovvio, a questo punto, che se noi inseriamo una lettera al posto di un numero, il computer ci segnala che non capisce cosa stiamo facendo e ci 'prega' di ripetere.

Sappiamo tutti, che lavorando sulla tastiera, è molto facile 'toccare' per sbaglio un tasto che non "c'entra"! Si può prevedere questo tipo di errore e corregerlo, sostituendo la variabile R con la variabile R$: cioè risponderemo con un carattere anzichè un numero. In questo modo, fra l'altro, avremo a disposizione (per la nostra programmazione) praticamente tutta la tastiera e non solo i 10 numeri (da 0 a 9).

Quindi, modifichiamo le ultime linee nel seguente modo:

```
200 INPUT R$
210 IF R$= "1" THEN 500
220 IF R$= "2" THEN 600
230 IF R$< > "1" AND R$ < > "2"
    THEN 200
```

(dopo il THEN, non c'è bisogno del GOTO)

L'unica differenza con le precedenti linee è che ora l'1 e il 2 sono tra virgolette (""): per quanto detto precedentemente, l'1 e 2, sono trattati, come caratteri alfanumerici e non più come semplici numeri. Quindi fatte le modifiche, potete tranquillamente premere tutti i tasti che volete senza "rovinare" nulla.

Sembrerebbe così, che anche la parte 'scelta' sia terminata: chi non si accontenta avrà notato però come l'istruzione INPUT sia in grado di rovinare la schermata del 'menù', riempiendo il video, in caso di errori, di punti di domanda.

Esitono, ovviamente, metodi per tenere lo schermo 'pulito' (essenzialmente consistono nel far tornare il cursore dell'INPUT sempre nella stessa posizione), ma, credetemi, non ne vale pena! Senza nulla togliere all'istruzione INPUT (che dà notevoli vantaggi in altri casi), penso sia meglio ricorrere, per scegliere un'opzione del menù, all'istruzione GET.

Provate subito questa linea:

```
1000 GET R$
```

e battete:

RUN 1000 (RET).

Vedrete comparire, istantaneamente, READY: sullo schermo non è comparso nessun punto interrogativo (?) e non c'è stata alcuna attesa; il programma, come è iniziato, è già finito! Ecco dunque una prima differenza. Quando il programma g ira e incontra un'istruzione GET, se in quel momento è stato premuto un tasto accetta la variabile, altrimente prosegue.

La tecnica per far attendere il computer fino alla pressione di un tasto, è la seguente:

```
1000 GET R$: IF R$T“” THEN 1000
```

Ora, se R$ coincide con un 'niente' (tra le virgolette, poste l'una dopo l'altra, non c'è alcun segno), il programma ritorna sulla linea stessa finchè non verrà premuto un tasto, permettendo al programma di proseguire.

Per il C 16, la linea 1000 può essere sostituita dalla seguente:

```
1000 GETKEY R$
```

Un'altra differenza tra INPUT e GET, è data dal fatto che Input accetta le variabili dopo che il tasto (RET) è stato premuto, quindi variabili costituite anche da più lettere (o numeri), mentre GET preleva direttamente i caratteri mentre vengono premuti (se si dovessero usare variabili di più caratteri, allora diventa più complesso l'uso di GET al posto di INPUT.

Visto che si sta parlando di differenze, ce n'è un'altra che consiste nel fatto che se volete uscire dal programma quando siete in un istruzione GET, basta premere il tasto RUN/STOP e il programma si ferma; se invece siete in un'istruzione INPUT, la pressione del solo tasto RUN/STOP non è sufficiente. Occorre premere, contemporaneamente, anche il tasto RESTORE.

Tornando alla nostra 'scelta', ecco come modificare le linee da 200 a 230 con

l'uso di GET:
200 GET R$
210 IF R$="1" THEN 500
220 IF R$="2" THEN 600
230 GOTO 200
(le linee 210 e 220 son sempre le stesse)

Nella linea 200 abbiamo sostituito il GET al posto dell'INPUT. Se le condizioni di 210 e 220 non sono verificate, il programma arriva in 230 e da qui ritorna a 200 finchè la pressione del tasto 'corretto' non lo farà uscire da questo 'LOOP'. Operando in questo modo, evitiamo anche i ciclo di attesa di linea 1000.

A questo punto, abbiamo visualizzato il menù e abbiamo la possibilità di scegliere l'opzione senza rovinare la schermata. Non ci resta che costruire le routines 'operative'.

Un altro modo per vedere ciò che abbiamo fatto finora, è quello di osservare il seguente schema che non è un programma, ma una schematizzazione del programma prima esaminato:

100 VISUALIZZA MENU'
:
:
200 SCELTA
210 SE 1 SALTA AL QUADRATO
220 SE 2 SALTA AL TRIANGOLO
230 SE NIENTE, OPPURE ERRORI, TORNA ALLA SCELTA
:
:
500 ESEGUE QUADRATO
510 RITORNA MENU'
:
600 ESEGUE TRIANGOLO
610 RITORNA MENU'

Tenendo presente tale schema, riscriverò, per comodità, tutto quello che è stato fatto finora. Voglio precisare una cosa: le linee 510 e 610, rimandano il programma all'inizio, in modo da poter continuare senza dover sempre ribattere RUN dopo ogni calcolo.:

Se R$="1", il programma salta alla linea 500 (routine Quadrato). La prima cosa che succede, è che si azzera lo schermo e compare la scritta "QUADRATO". La linea 510 ci rimanda al menù. La linea 505 serve come 'interruttore' tra 500 e 510: il programma non passa in 510 finchè non è stato premuto un tasto qualsiasi. Questo 'interruttore' è stato costruito come la linea 1000 vista in precedenza: è indispensabile agire così perchè, una volta operata la scelta da menù, il programma salta al quadrato oppure al triangolo e senza interruttore, ritorna immediatamente al menù.

Provate, comunque, a cancellare le linee 505 e 605 e osservate la differenza.

Di seguito ricordiamo che per cancellare una linea BASIC è sufficiente digitare il suo numero e premere il tasto RETURN.

Routines quadrato e triangolo

Occupiamoci ora di come risolvere il problema del calcolo delle aree.

E' necessario predisporre un INPUT che 'riceva' il valore del lato (per entrambe le routines). Se guardate la Fig. 2, vedrete come il problema è stato risolto. E' bene tener presente che alle righe del listato di figura 2 vanno aggiunte le righe da 100 a 230 del programma finora visto. La maggior parte delle istruz ioni sono di carattere 'frivolo': servono solo a centrare i vari messaggi ed a rendere la schermata elegante (!). Una volta apparso il risultato, per non farlo scomparire immediatamente col ritorno al menù, ho inserito 'l'interruttore' di linea 505 (che ora diventa 570). Osservate la linea 520 (o 630), che riporto per comodità:
520 INPUT "LATO "

Come avete notato, in una istruzione di INPUT, si può inserire anche un messaggio, basta che questo sia trascritto all'in-

```
450 REM  ****************
460 REM  * MENU': FIG. 2 *
470 REM  ****************
480 :
490 REM  *** ROUTINE QUADRATO ***
500 PRINTCHR$(147)CHR$(17)SPC(12)"
    QUADRATO"
510 PRINTCHR$(17)CHR$(17)SPC(5);
520 INPUT "LATO ";L
530 AR=L*L
540 PRINTCHR$(17)CHR$(17)CHR$(17)S
    PC(5)"L'AREA E' :"AR
550 FOR I=0 TO 5:PRINTCHR$(17):NEX
    T
560 PRINTSPC(10)"PREMI UN TASTO"
570 GET A$:IF A$="" THEN 570
580 GOTO 100
590 :
600 REM  *** ROUTINE TRIANGOLO ***
610 PRINTCHR$(147)CHR$(17)SPC(12)"
    TRIAN.EQ"
620 PRINTCHR$(17)CHR$(17)SPC(5);
630 INPUT "LATO ";L
640 AR=L*L*SQR(3)/4
650 PRINTCHR$(17)CHR$(17)CHR$(17)S
    PC(5)"L'AREA E' :" AR
660 FOR I=0 TO 5:PRINTCHR$(17):NEX
    T
670 PRINTSPC(10)"PREMI UN TASTO"
680 GET A$:IF A$="" THEN 680
690 GOTO 100
```
Fig. 2

terno di virgolette (") e seguito da un punto e virgola (;), altrimenti il computer segnala un errore di sintassi.

Quello che succede è che ora il punto di domanda (?) tipico dell'input, comparirà di fianco alla parola LATO.

Inoltre la variabile rappresentante il lato è indicata con la lettera L, lettera, che caratterizza le variabili numeriche (manca il '$' delle alfanumeriche): quindi il computer si aspetterà un numero. Cosa c'è di strano, direte voi, dato che il valore del lato da inserire è un numero? In effetti non c'è niente di strano, salvo che così facendo, non teniamo conto del fatto che la nostre mani potrebbero cadere dovunque sulla tastiera, anche su 'lettere', con il castigo della solita sgridata 'REDO FROM START', e conseguente 'abbruttimento' della videata.

Il tutto può essere superato, come visto in precedenza, con l'uso di una variabile Alfanumerica, tipo 'L$' la quale, però, dopo l'INPUT, andrebbe ritrasformata in numero.

Visto che quello che abbiamo finora non è poco, preferisco rimandare il discorso sull'INPUT 'controllato' ad un altro numero della rivista.

Infine, se dovete inserire un valore del lato che contiene decimali, ricordatevi di usare il PUNTO al posto della VIRGOLA: se volete comunque provare ad usare la (,), come: 5,3 otterrete il messaggio 'EXTRA IGNORED', il che vuol dire che i numeri dopo la virgola, sono stati ignorati (è stato accettato solo il 5, ed il risultato sarà un'area di 25).

Per semplicità, vorrei ora riportare la schema delle due routines:

A1 STAMPA 'QUADRATO'
A2 CHIEDE VALORE LATO
A3 CALCOLA AREA
A4 STAMPA RISULTATO
A5 ASPETTA LA PRESSIONE DI UN TASTO

:

:

B1 STAMPA 'TRIAN.EQ'
B2 CHIEDE VALORE LATO
B3 CALCOLA AREA
B4 STAMPA RISULTATO
B5 ASPETTA LA PRESSIONE DI UN TASTO

Potete accorgervi facilmente come alcune parti delle due routines siano identiche. Questo suggerisce l'idea di creare delle Subroutine che si occupino di svolgere quelle determinate mansioni, ritornando poi alle routines principali.

Una Subroutine è un sottoprogramma che viene richiamato dall'istruzione GOSUB xxx. Quando il programma principale incontra un'istruzione GOSUB, 'prende nota' della posizione di partenza e poi salta alla Subroutine; segue poi quello che la subroutine stessa deve fare e ritorna al punto in cui aveva lasciato il programma principale. Perchè ciò avvenga correttamente, la subroutine deve terminare con l'istruzione RETURN. Si

```
100 REM  *****************
110 REM  * MENU': FIG. 3 *
120 REM  *****************
130 :
140 REM  *** VISUALIZZA MENU' ***
150 PRINTCHR$(147)CHR$(17)CHR$(17)
    CHR$(17)
160 PRINTSPC(12)"QUADRATO   "CHR$(1
    8)" 1 "CHR$(17)
170 PRINTSPC(12)"TRIAN.EQ.  "CHR$(1
    8)" 2 "CHR$(17)
180 :
190 REM  *** SCELTA ***
200 GET R$
210 IF R$="1" THEN 260
220 IF R$="2" THEN 300
230 GOTO 200
240 :
250 REM  *** QUADRATO ***
260 PRINTCHR$(147)CHR$(17)SPC(12)"
    QUADRATO":GOSUB 340
270 AR=L*L:GOSUB 380:GOTO 150
280 :
290 REM  *** TRIANGOLO ***
300 PRINTCHR$(147)CHR$(17)SPC(12)"
    TRIAN.EQ":GOSUB 340
310 AR=L*L*SQR(3)/4:GOSUB 380:GOTO
     150
320 :
330 REM  *** INPUT SUBROUTINE ***
340 PRINTCHR$(17)CHR$(17)SPC(5)
350 INPUT "LATO ";L:RETURN
360 :
370 REM  *** RISULTATO ***
380 PRINTCHR$(17)CHR$(17)CHR$(17)S
    PC(5)"L'AREA E' :"AR
390 FOR I=0 TO 5:PRINTCHR$(17):NEX
    T
400 PRINTSPC(10)"PREMI UN TASTO"
410 GET A$:IF A$="" THEN 410
420 RETURN
```

Fig.3

possono anche creare Subroutine all'interno di altri sottoprogrammi, però dovete sempre ricordarvi di 'chiudere' ognuna di esse con RETURN. Se richiamate più sottoprogrammi e vi dimenticate di chiuderli, potete incappare in un errore del tipo 'OUT OF MEMORY' (non c'è più memoria disponibile). Quindi, per evitare il ripetersi di istruzioni identiche, possiamo far us o di Subroutines che si occupino dell'INPUT, della stampa del Risultato o della pressione di un tasto.
Lo schema andrà modificato in questo modo:

A1 STAMPA 'QUADRATO'
A2 GOSUB C1
A3 CALCOLA AREA
A4 GOSUB C2
A5 RITORNA MENU'
:
B1 STAMPA 'TRIAN.EQ'
B2 GOSUB C1
B3 CALCOLA AREA
B4 GOSUB C2
B5 RITORNA MENU'
:
C1 INPUT SUBROUTINE - RETURN
:
C2 STAMPA AREA - ASPETTA PRESSIONE TASTO - RETURN

Il tutto è realizzato nel programma di FIGURA 3, che può essere ritenuto conclusivo, anche se è possibile andare oltre. Per esempio, supponiamo di voler calcolare l'area di due quadrati.

Col nostro programma calcoleremmo l'area del primo. In seguito dovremmo tornare al menù, scegliere nuovamente l'opzione 'quadrato' (premendo il tasto 1), e calcolare la seconda area. Ora, sarebbe comodo poter calcolare la seconda area senza ripassare dal menù, cioè ritornare all'inizio della routine quadrato (o triangolo); lascio a voi il compito in modo che vi possiate 'esercitare'.

Di seguito come aiuto, potete osservare il diagramma a blocchi di fig.4.

Infine un'ultimo esercizio: provate a prevedere l'uso di un tasto che vi faccia uscire definitivamente dal programma!.

Giancarlo Castagna

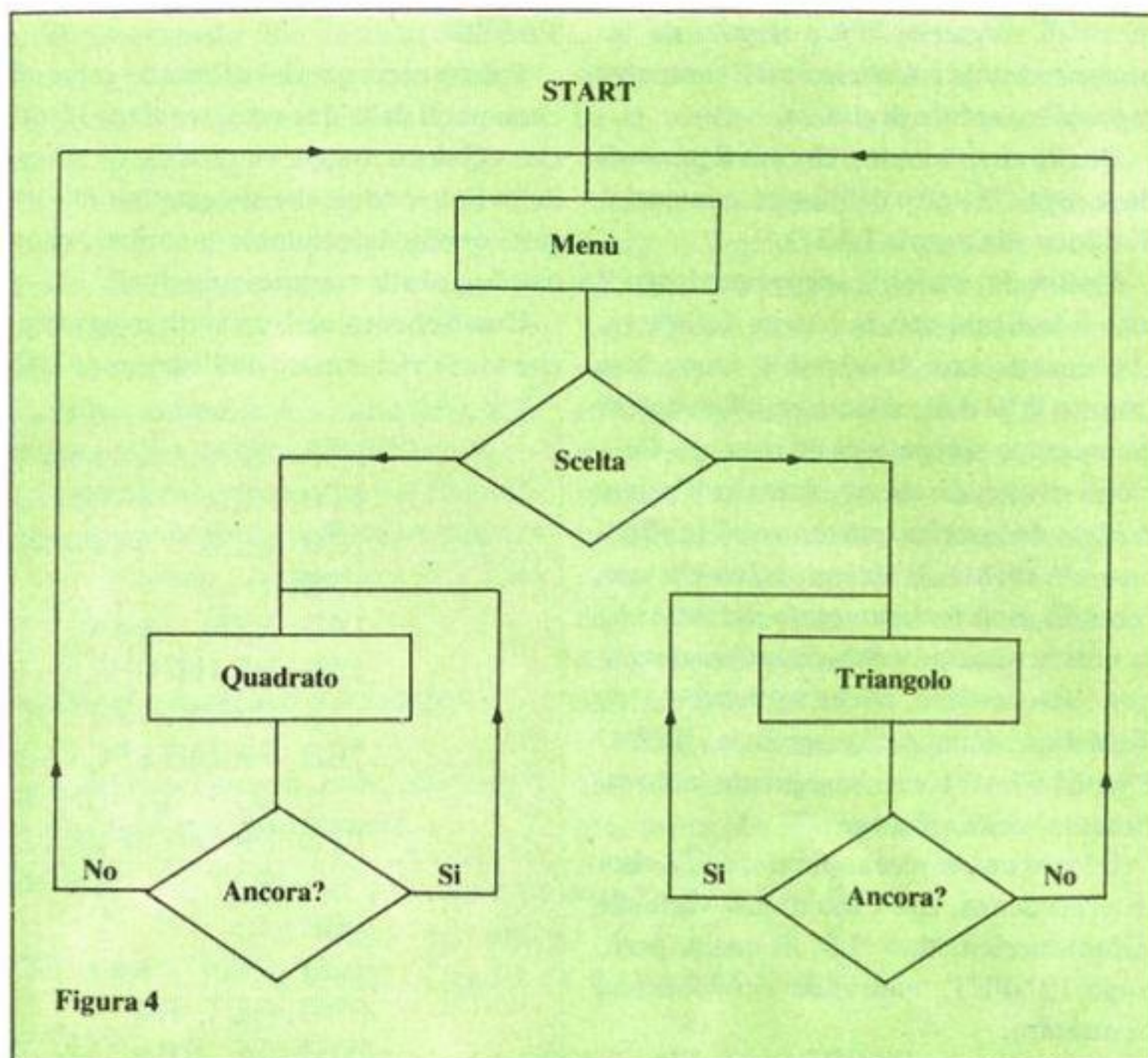
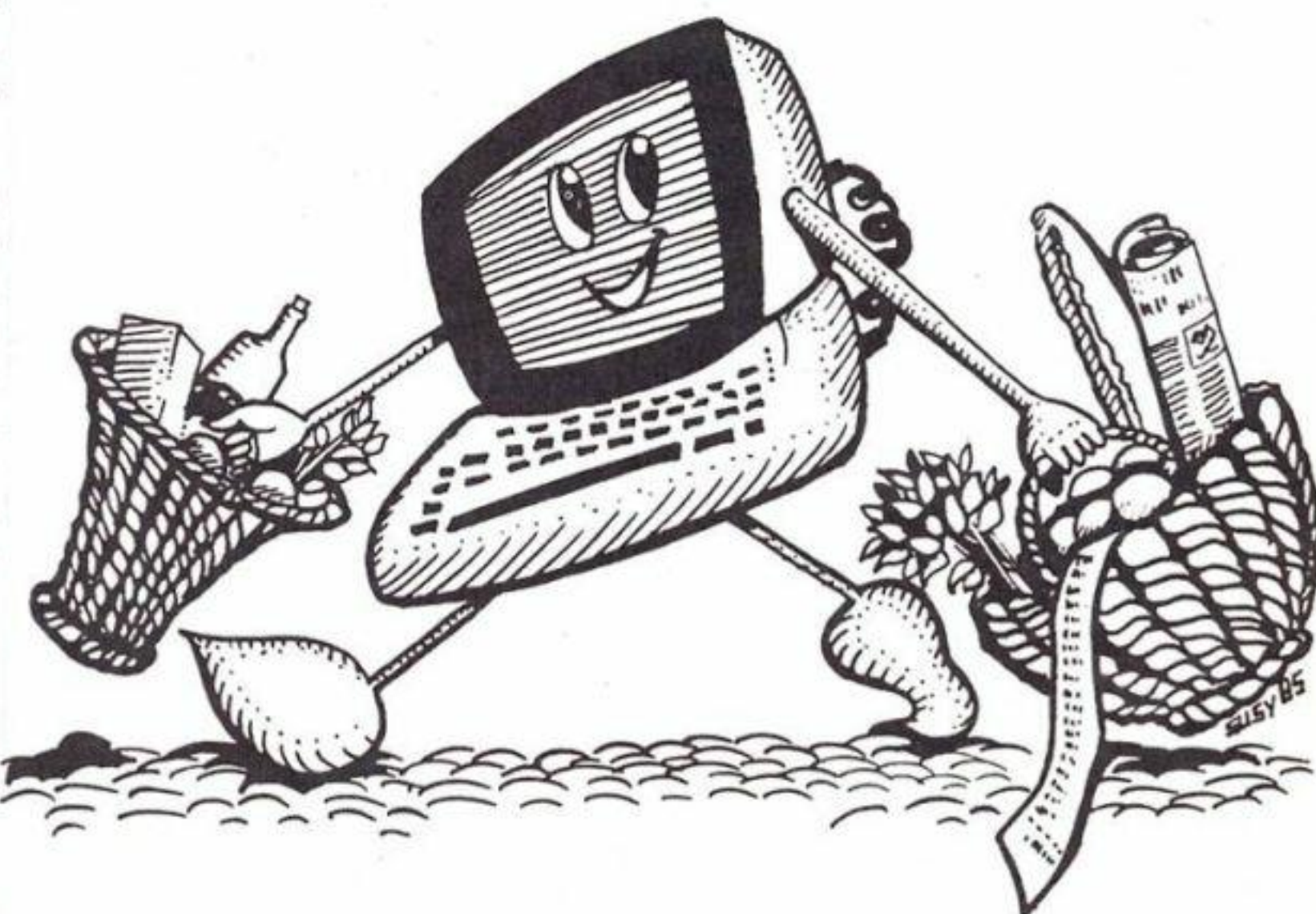



Figura 4

MONITOR LM DEL C16

Le potenti istruzioni incorporate nelle Rom dell'ultimo computer Commodore.

Leggendo il manuale d'uso che accompagna il C16 ci si accorge facilmente di piccole imperfezioni sparse qua e là. La più grave è senza dubbio quella sul comando del nuovo Basic 3.5 : Monitor.

Viene detto che tale comando ci porta fuori dall'"ambiente" Basic e consente di entrare nel programma monitor per correggere e/o scrivere programmi in linguaggio macchina. Dopodichè troviamo testualmente scritto: 'Vedere la sezione sui comandi del monitor per maggiori informazioni'. Dov'è questa sezione? Non c'è.

Ecco allora che Commodore Computer Club ha pensato di venire in aiuto ai possessori di questo piccolo, ma potente computer.

Il programma Monitor che è incorporato sul C16, detto TEDMON dal nome dell'integrato video (TExt Display chip), comprende, oltre al monitor per il linguaggio macchina, un miniassembler ed un minidisassembler. Vediamo ora i suoi comandi.

I comandi

A = assembly: permette di assemblare le istruzioni mnemoniche dello standard MOS, quali: LDX, LDA, BIT, PHP, ecc.

Sintassi: A indirizzo cod. operativo operando dove "indirizzo" è un numero esadecimale che si riferisce alla locazione di memoria in cui assemblare l'istruzione.

Premendo <RETURN> dopo aver scritto l'istruzione si passa automaticamente alla locazione di memoria successiva. Per uscire dall'assemblaggio è sufficiente premere il tasto <RETURN> senza aver scritto alcuna istruzione.

Esempio:
A 1100 LDA $00

C = compare: confronta due aree di memoria e segnala le eventuali differenze stampando sul video le locazioni di memoria relative.

Sintassi: C inizio1area fine1area inizio2area

Anche in questo caso i parametri "inizio1area", "fine1area" ed "inizio2area" sono numeri esadecimali che indicano le locazioni di memoria in esame.

Esempio:
C 1000 2000 3000

D = disassembly: disassembla il linguaggio macchina in mnemonico.

Sintassi: D inizioarea finearea con gli indirizzi sempre in esadecimale.

Occorre notare che se omettiamo l'indirizzo di fine vengono mostrate 20 istruzioni dopodichè si ha il ritorno del cursore lampeggiante.

Premendo D e <RETURN> si ha la possibilità di vedere le 20 istruzioni successive. Se dopo aver disassemblato un'area di memoria ci posizioniamo con il cursore sopra una istruzione e la modifichiamo, premendo <RETURN> si entra automaticamente in modo A)ssembly.

Esempio:
D 1000 1020
oppure
D 1000

F = fill: riempie un'area di memoria con uno specifico valore di un Byte.

Sintassi: F inizioarea finearea valore anche il valore come gli indirizzi che indicano le aree deve essere in esadecimale.

Esempio:
F 1000 2000 EA

G = GOTO permette di mandare in esecuzione un programma ad un indirizzo specificato

Sintassi: G indirizzo

Se si omette l'indirizzo l'esecuzione sarà eseguita a partire dal corrente valore del Program Counter (vedere comando R).

Esempio:
G 1200

H = hunt: ricerca dei dati specificati in una determinata area.

Sintassi: H inizioarea finearea dati
i dati possono essere sia valori esadecimali, che stringhe alfanumeriche. Le stringhe occorre siano precedute dall'apostrofo (ottenibile con <SHIFT+7>). Quando i dati vengono trovati entro l'area in esame si ha la stampa su video degli indirizzi ricercati.

Esempio:
H 1000 2000 20 D2 FF
oppure
H E000 FFFF 'commodore

L = load: carica da una memoria di massa un file specifico.

Occorre notare che il programma caricato in memoria sia da nastro che da disco, non viene mai rilocato. Si posiziona cioè nella stessa area di memoria da cui era stato salvato.

Sintassi: L nomeprogramma, numeroperiferica

Il nome del programma va scritto tra virgolette (come in Basic) mentre il numero della periferica deve essere in esadecimale.

sempio:
L "PROVA C16", 01 (carica da nastro)
oppure
L "PROVA C16", 08 (carica da disco)

M = memory: visualizza il contenuto di un'area di memoria con una stampa su video dei codici esadecimali e ASCII.

Sintassi: M inizioarea finearea
Come per il comando D anche in questo caso se si omette l'indirizzo di finearea saranno visualizzate 96 locazioni di memoria (12 righe di 8 byte alla volta) e premendo M seguito da <RETURN> si potranno vedere le 96 locazioni successive.

Si possono cambiare i valori dei byte esadecimali e confermare la modifica con il <RETURN>. In caso di errore, causato o da un tentativo di scrivere sulla ROM o da un codice inesistente, si ha la visualizzazione di un punto interrogativo subito dopo la locazione errata.

Esempio:
M 1000 2000
oppure
M 1000

R = register: visualizza i registri vitali del microprocessore 7501 (non spaventatevi anche se ha una sigla diversa in pratica è come il famoso 6502).

In particolare vengono mostrati:
il PC, cioè il contatore di programma
lo SR, cioè il registro di Stack
lo AC, cioè l'accumulatore
il XR, cioè il registro X
il YR, cioè il registro Y
lo SP, cioè il puntatore allo Stack

Sintassi: R ovviamente seguito dal tasto <RETURN>.

E' interessante notare il fatto che è possibile variare il contenuto dei vari registri nello stesso modo usato per variare il contenuto delle locazioni di memoria.

S = salvare: salva il contenuto di un'area di memoria specificata.

Sintassi: S nomefile, nem. periferica, inizioarea, finearea

Occorre notare che l'indirizzo di finearea va aumentato di 1 rispetto all'area dati che vogliamo salvare.

Esempio:
S "C16 PROVA",01,1000,2001
(salva l'area da $1000 a $2000 su nastro)
oppure
S "C16 PROVA" ,08,3500,4000
(salva su disco l'area da $3500 a $3fff)

T = transfert: ricopia un'area di memoria in un'altra area specificata.

Sintassi: T inizio1area fine1area inizio2area dove 2area è quella dove andrà a riprodursi l'area.

Esempio:
T 1000 2000 3000
oppure
T 5000 5500 1000

V = verify: confronta un file su nastro o su disco con l'area di memoria in cui il file sarebbe caricato. Se si ha una differenza viene visualizzata la scritta VERIFYING ERROR.

Sintassi: V nomefile, nem. periferica

Esempio:
V "C16 PROVA",01
(da nastro) oppure
V "C16 PROVA",08
(da disco)

X = exit: esce dall'ambiente Monitor e torna al Basic.
E' buona norma, una volta tornati al Basic, di dare il comando CLR che permette di inizializzare il puntatore allo Stack..

Sintassi: X seguito dal <RETURN>.

Memoria e locazioni

Adesso che siamo in grado di utilizzare al meglio il programma TEDMON vediamo di dare una sbirciatina alla memoria del C16. A grandi linee la mappa di memoria del C16 è organizzata come in

ESA/DEC	NOTE
$0000/0 $07FF/2047	VARIABILI DI SISTEMA
$0800/2048 $0BFF/3071	AREA RAM COLORE SCHERMO
$0C00/3072 $0FFF/4095	AREA RAM SCHERMO
$1000/4096	INIZIO AREA RAM BASIC
$1800/6144 $1BFF/7167	LUMINANZA BIT DELLA MAPPA SCHERMO
$1C00/7168 $1FFF/8191	MAPPA COLORE SCHERMO IN ALTA RISOLUZIONE
$2000/8192 $3FFF/16383	AREA RAM SCHERMO HI-RES OPPURE AREA BASIC
$8000/32768 $BFFF/49151	INTERPRETE BASIC
$D000/53248 $D7FF/55295	ROM CARATTERI
$D800/55296 $FFFF/65535	ROM SISTEMA OPERATIVO

figura 1.

Analizziamo ora più da vicino alcune locazioni che possono risultare interessanti per determinate situazioni.

Innanzitutto i vari puntatori del Basic sono rimasti i medesimi del VIC 20 e del CBM 64 e cioè:
$2b-2c/43-44 = inizio area Basic
$2d-2e/45-46 = fine area Basic e inizio area variabili
$2f-30/47-48 = inizio area Arrays
$31-32/40-50 = fine area Arrays
$33-34/51-52 = fine area stringhe (N.B. le stringhe vengono memorizzate a partire dalla fine della memoria e andando a ritroso).

Altre utili locazioni possono essere:
$00ca/202 = numero colonna in cui si trova il cursore.
$00cd/205 =numero riga in cui si trova il cursore.

E' da notare il fatto che se scriviamo: POKE 202,2: POKE 205,10 sia in modo diretto che in programma, avremmo il risultato di posizionare il cursore alla seconda posizione della riga decima. Avremmo così ottenuto il risultato di un'istruzione del tipo PRINT AT non disponibile nel Basic dei Commodore.
$0540/1344 = Flag di AUTOREPEAT.

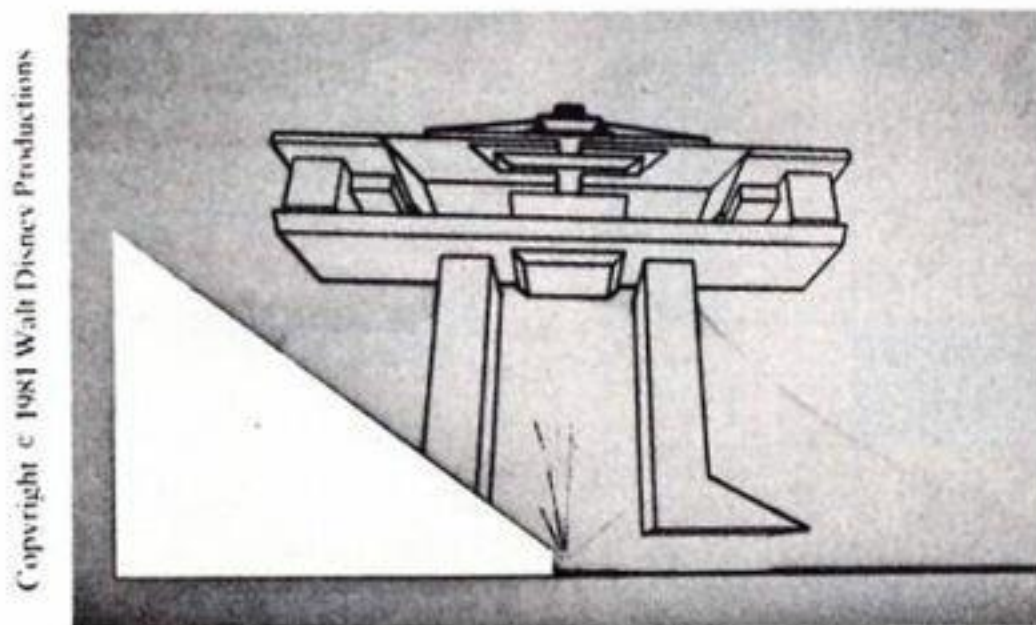
Il C16 ha già la ripetizione su tutti i tasti e questo a volte può non essere desiderato. Allora ecco che conoscendo questa locazione di memoria è possibile disabilitare la ripetizione dei tasti (a parte la barra spaziatrice, il Delete, ecc.) I valori da immettere in questa locazione sono: 128 per avere la ripetizione su tutti i tasti e 0 per disabilitarla.
$0543/1347 = Flag per il controllo dei tasti SHIFT, CTRL e COMMODORE.

I valori precisi sono:
1 -> Tasto SHIFT
2 -> Tasto ÇOMMODORE (in basso a sinistra)
4 -> Tasto CTRL

Una utilizzazione di tale locazione può essere quella di attendere che sia premuto uno di questi tasti per andare avanti nell'esecuzione del programma.

Esempio: 30 WAIT 1347,1
attende fino a che non si preme il tasto SHIFT.
$00ef/239 = numero di caratteri nel buffer di tastiera.

Permette di forzare i caratteri nel Buffer (posto da $0527/1319 a $0530/1328) e alla fine di scaricarli su video. E' un trucchetto molto spesso utilizzato per varie situazioni (vedi DATA MAKER per CBM 64 e VIC 20).

Giovanni Beani

La navicella del film Tron, sulla quale viaggiava il protagonista, danneggiata dopo un incidente.

di **Giancarlo Castagna**

Marzio, dopo un lungo periodo di 'letargo' e dopo essersi visto per l'ennesima volta allo specchio, decide di fare un pò di sport. Pensando che nel periodo estivo passa sempre qualche giorno in montagna e che gli piacerebbe fare qualche arrampicata un po' seria (con l'aiuto di una guida, ovviamente), decide di iniziare i propri allenamenti su una specie di spalliera svedese. Lo scopo che si prefigge è quello di percorrere più volte il 'quadro', lasciando dietro di sè delle palline, in modo da capire quando l'intero percorso è stato ultimato.

Purtroppo, si accorgerà presto che le cose sono più complicate di quello che pensava! Difatti, sullo stesso percorso, incontrerà altre persone non proprio educate, che continueranno il loro allenamento senza preoccuparsi della sua presenza. Sta quindi a lui cercare di evitarle!

Sia bene inteso che nessuno insegue Marzio, ma che tutti, prima o poi, se non starà più che attento, si scontreranno con lui, trascinandolo 'fuori pista'.

Inoltre Marzio ha a disposizione solo tre prove.

Come si gioca

Scopo del gioco è guidare Marzio (sprite nero, inizialmente in basso a destra) lungo il percorso, con l'aiuto del Joystick in port 2.

I quadri da completare sono quattro, dopodichè si proseguirà sempre con la configurazione del quarto.

Gli avversari da evitare vanno in ordine crescente con il numero del quadro (primo quadro 1, secondo 2, terzo 3 e quarto 4).

Tutti gli sprite 'nemici', seguono percorsi prestabiliti a voi sconosciuti: quando li avrete scoperti, alla vostra abilità si dovrà aggiungere anche una certa strategia per riuscire a terminare il percorso senza subire danni!

Alla fine del gioco, premendo il tasto Y si ricomimcia, prememdo N si esce.

Nel digitare il programma, non è necessario battere le linee REM, in quanto ad esse non si fa riferimento.

Un "percorso di guerra" per riprendere la forma prima delle ferie.

Il programma

Il Problema da affrontare è quello di far muovere lo sprite (Marzio) su un percorso stabilito, facendogli lasciare, contemporaneamente, una scia.

Se qualcuno ha letto gli articoli pubblicati sui numeri 17 e 18 della rivista riguardanti la gestione degli sprite, saprà che esistono due registri che si occupano della collisione tra sprite e della collisione tra sprite e fondo.

Ripetiamo comunque, per chi non lo ricordasse, alcune informazioni di carattere generale. I due registri sono:

53278 collisione tra sprite
53279 collisione tra sprite e fondo

Il registro 53278 è quello che fa per noi, in quanto permette di determinare la collisione tra Marzio ed un qualsiasi altro sprite. Il secondo registro invece non è assolutamemte in grado di capire qual è il carattere entrato in collisione con lo sprite e non ci è di alcuna utilità.

Per far muovere Marzio su un percorso prestabilito, è necessario sapere qual è il carattere che sta sotto di esso e, per fare questo, occorre sapere in che locazione dello schermo si trova il carattere stesso (dopodichè, basterà una semplice PEEK...)

Per un gioco del genere lo sprite si deve muovere di 8 pixel alla volta, sia in orizzontale che in verticale (difatti, un qualsiasi carattere è fatto da una matrice di 8 x 8 pixel).

Il tutto è stato risolto in questo modo: una volta costruito il 'quadro', si centra la 'pancia' di Marzio nella posizione di partenza e si calcola la locazione di schermo proprio sotto la pancia (1931). Per fare questo, può essere utile riportarsi lo schema del gioco su un foglio a quadretti: ogni quadretto, è una locazione di schermo.

Quando col Joystick si sposta lo sprite verso sinistra (o verso destra), per vedere qual è il carattere X-esimo sotto di esso, basta decrementare (o incrementare) la locazione iniziale di 1 e leggere, con una PEEK(X), il suo contenuto. Se invece lo sprite si sposta verso l'alto (o verso il basso), occorre decrementare (o incremetare) la locazione iniziale di 40.

Nel listato, il procedimento sopra descritto è fatto nelle linee da 510 a 540, le quali, con un GOTO e un GOSUB, fanno saltare il programma alle linee 1500 - 1530. Queste ultime si occupano di controllare il carattere al di sotto della 'pancià di Marzio.

Per muovere lo sprite sul percorso del gioco non occorre, in effetti, sapere qual è l'esatto carattere sotto di esso: quello che ho fatto, è stato controllare semplicemente che non si tratti del "vuoto" CHR$(32). Se si tratta proprio di questo, si pone una variabile K uguale ad uno e, nelle linee 1010 - 1110 - 1210 - 1310, si annullano gli incrementi (o decrementi) sia delle locazioni di schermo che delle coordinate dello sprite, in modo che lo sprite stesso non subisca spostamenti.

Nella linea 1510, oltre a controllare che il carattere sotto lo sprite non sia il 32, si controlla anche che non sia l'81

(CHR$ corrispondente al pallino pieno: rappresenta la "scia").

Tenete comunque presente che gestire 5 sprite in Basic comporta una certa lentezza: il metodo descritto sopra, è quello che, a mio avviso, 'lumachizza' al minimo (nei limiti del basic), i vari movimenti. Per quanto riguarda il 'resto' del programma, non dovrebbero esserci grossi problemi. Per facilitare comunque la lettura, ecco la lista di tutte le variabili con il loro valore iniziale:

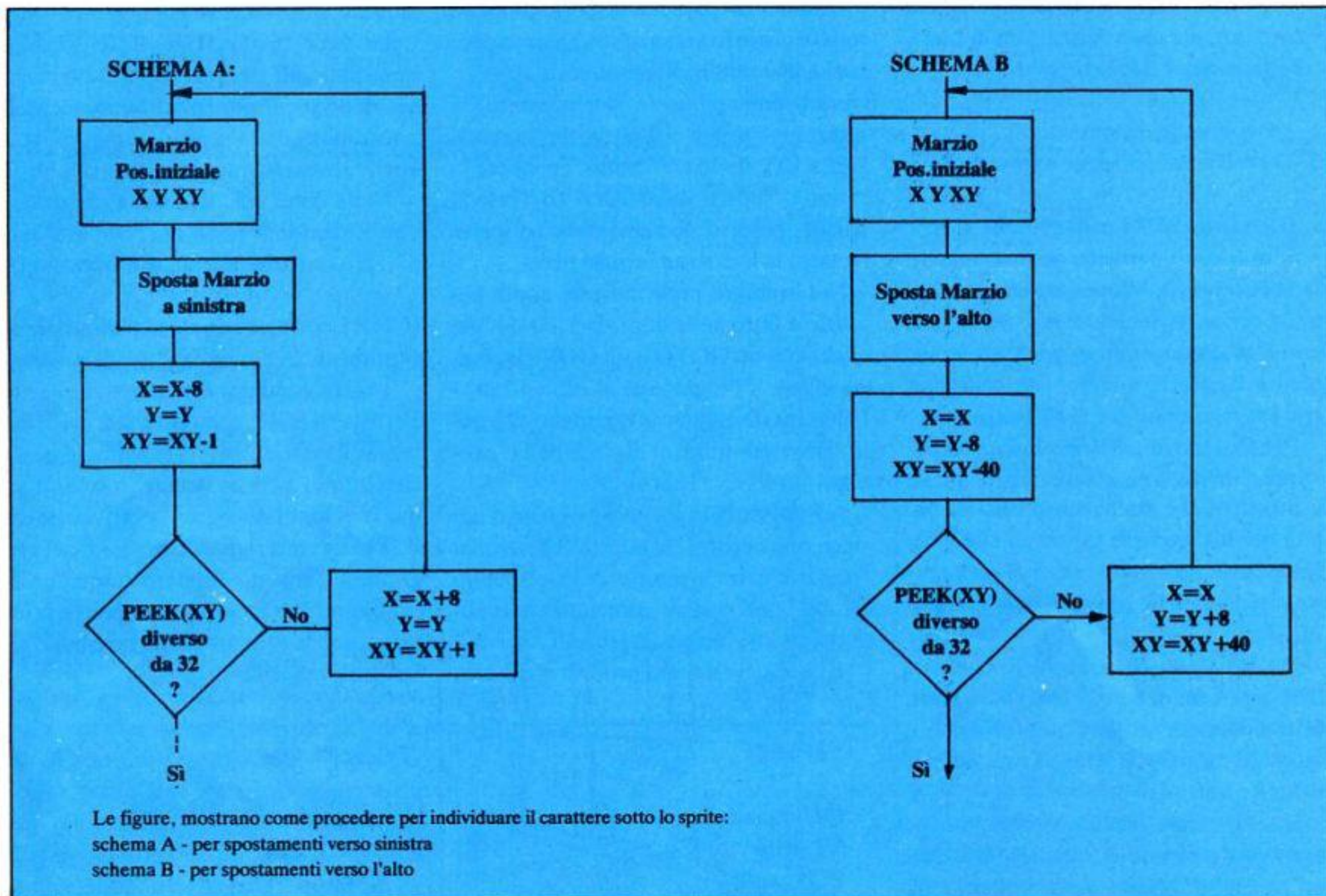
Variabili	Valore iniziale
QU : quadro	1
VI : vite	3
PU : punti	0
PA : punti fine quadro	0
XY : locazione schermo	1931
X,Y : coord. Marzio	vedi linea 400 - 410
X1,Y1;X2,Y2 : coord. sprite	vedi linee 400 - 410
X3,Y3;X4,Y4 : coord. sprite	vedi linee 400 - 410

GIOCHI

Le soluzioni

1 = I; 2 = B; 3 = F; 4 = D; 5 = E;
6 = C; 7 = G; 8 = H; 9 = A;
10 = J; 11 = P; 12 = L; 13 = M;
14 = Q; 15 = O; 16 = K; 17 = N;
18 = R; 19 = S; 20 = £; 21 = U;
22 = V; 23 = Z; 24 = X; 25 = Y;
26 = W; 27 = &; 28 = T.

■	■	■	1 L	E	N	■	2 C	O	3 L	O	R
■	4 P	E	E	K	■	■	■	5 L	E	■	■
■	6 O	U	T	■	■	7 R	E	A	D	Y	■
8 O	K	■	■	9 N	10 O	T	■	■	■	■	■
11 N	E	X	12 T	■	R	■	13 G	O	■	14 M	15 L
■	■	■	O	■	■	16 R	O	M	■	■	O
17 I	18 F	■	19 P	20 R	I	N	T	■	21 C	I	A
■	22 O	C	■	U	■	23 D	O	■	24 S	I	D

Le figure, mostrano come procedere per individuare il carattere sotto lo sprite:
schema A - per spostamenti verso sinistra
schema B - per spostamenti verso l'alto

```
1 REM   ************************
2 REM   * COMMODORE 64         *
3 REM   *                      *
4 REM   *       MARZIO         *
5 REM   *                      *
6 REM   *  GIANCARLO CASTAGNA  *
7 REM   ************************
8 :
9 :
100 PRINT"[CLEAR][DOWN]":VI=3:QU=1
    :GOSUB 2000
110 FOR I=0 TO 62:READ A:POKE 1228
    8+I,A:POKE 12352+I,A:POKE 1241
    6+I,A
120 CK=CK+A:NEXT
130 IF CK<>5343 THEN PRINT"ERRORE
    NEI DATA SPRITE 1 2 3":END
140 FOR I=0 TO 62:READ B:POKE 1248
    0+I,B:POKE 12544+I,B:CH=CH+B:N
    EXT
150 IF CH<>5814 THEN PRINT"ERRORE
    NEI DATA SPRITE 4 5":END
193 :
200 A1$="[ARANC]    ├────┼────┼─
    ──┼────┤": REM  [C]+Q;[SH]+*
    ;[SH]++;[C]+W
210 A2$="[ARANC]    │    │    │
    │    │": REM  [SH]+B
220 A3$="   │    │    │    │
    │": REM  [SH]+B
230 A4$="   └────┴────┴────┴─
    ──┘[CELESTE]": REM  [C]+Z;[SH]
    +*;[C]+E;[C]+X
240 PRINT"[CLEAR][DOWN]":POKE 5328
    0,11:POKE 53281,11

297 REM   ********************
298 REM   * PROGR.PRINCIPALE *
299 REM   ********************
300 FOR I=1 TO 3:PRINTA2$:PRINTA3$
    :PRINTA1$:PRINTA2$:PRINTA3$
310 PRINTA1$:NEXT:PRINTA2$:PRINTA3
    $:PRINTA4$
320 PRINT"[HOME][2 DOWN]"SPC(30)"[
    NERO] QUADRO-[CELESTE]":GOSUB
    3600
```

```
330 PRINT"[3 DOWN]"SPC(31)"[NERO]
    VITE-[CELESTE]":GOSUB 3500
340 PRINT"[3 DOWN]"SPC(31)"[NERO]
    PUNTI[CELESTE]"
350 V=53248:POKE 2040,192:POKE V+3
    9,0
360 POKE 2041,193:POKE V+40,3:POKE
     2042,194:POKE V+41,7
380 POKE 2043,195:POKE V+42,5:POKE
     2044,196:POKE V+43,8
400 X=232:Y=218:X1=40:Y1=58:X2=232
    :XY=1931
410 Y2=72:X3=232:Y3=58:X4=40:Y4=21
    8
420 IF QU=1 THEN POKE V+21,3:GOTO
    460
430 IF QU=2 THEN POKE V+21,7:GOTO
    460
440 IF QU=3 THEN POKE V+21,15:GOTO
     460
450 IF QU=>4 THEN POKE V+21,31
460 POKE V,X:POKE V+1,Y:POKE V+2,X
    1:POKE V+3,Y1
470 POKE V+4,X2:POKE V+5,Y2:POKE V
    +6,X3:POKE V+7,Y3:POKE V+8,X4:
    POKE V+9,Y4
480 P1=PEEK(53278)
499 :
500 P=PEEK(56320)
510 IF P=123 THEN X=X-8:XY=XY-1:GO
    TO 1000
520 IF P=119 THEN X=X+8:XY=XY+1:GO
    TO 1100
530 IF P=126 THEN Y=Y-8:XY=XY-40:G
    OTO 1200
540 IF P=125 THEN Y=Y+8:XY=XY+40:G
    OTO 1300
545 GOTO 560
550 POKE XY,81:POKE V,X:POKE V+1,Y
560 Y1=Y1+8:IF Y1>218 THEN Y1=58:X
    1=X1+48
570 IF X1>232 THEN X1=40
580 POKE V+2,X1:POKE V+3,Y1:GOSUB
    800
600 IF QU=1 THEN 770
610 X2=X2-8:IF X2<40 THEN X2=232:Y
    2=Y2+24
620 IF Y2>218 THEN Y2=72
630 POKE V+4,X2:POKE V+5,Y2:GOSUB
    800
650 IF QU=2 THEN 770
660 Y3=Y3+16:IF Y3>218 THEN Y3=58:
    X3=X3-48
670 IF X3<40 THEN X3=232
680 POKE V+6,X3:POKE V+7,Y3:GOSUB
    800
700 IF QU=3 THEN 770
720 X4=X4+16:IF X4>232 THEN X4=40:
    Y4=Y4-24
730 IF Y4<72 THEN Y4=218
750 POKE V+8,X4:POKE V+9,Y4:GOSUB
    800
770 IF PA=245 THEN 4000
780 GOTO 500
800 P1=PEEK(53278)
810 IF P1=3 THEN D=0:GOTO 3000
820 IF P1=5 THEN D=0:GOTO 3100
830 IF P1=9 OR P1=17 THEN 3200
840 RETURN
998 :
999 :
1000 GOSUB 1500:IF K<>1 THEN 550
1010 K=0:XY=XY+1:X=X+8:GOTO 550
1099 :
1100 GOSUB 1500:IF K<>1 THEN 550
1110 K=0:XY=XY-1:X=X-8:GOTO 550
1199 :
1200 GOSUB 1500:IF K<>1 THEN 550
1210 K=0:XY=XY+40:Y=Y+8:GOTO 550
1299 :
1300 GOSUB 1500:IF K<>1 THEN 550
1310 K=0:XY=XY-40:Y=Y-8:GOTO 550
1499 :
1500 PE=PEEK(XY):IF PE=32 THEN K=1:
     RETURN
1510 IF PE<>32 AND PE<>81 THEN PA=P
     A+1:PU=PU+1
1530 GOSUB 3700:RETURN
1999 :
2000 POKE 53280,0:POKE 53281,0
2050 PRINT"[HOME][5 DOWN]"SPC(14)"M
      A R Z I O"
2060 PRINT"[3 DOWN]"SPC(12)"JOISTIC
     K PORT 2"
2070 PRINT"[4 DOWN]"SPC(15)"UN ATTI
     MO"
2080 PRINT"[3 DOWN]"SPC(14)"@1985
     CAST"
2090 RETURN
2997 REM     ********************
2998 REM     * COLLIS. SPRITE 2 *
2999 REM     ********************
3000 Y1=Y1+8:GOSUB 4600
3010 POKE V+2,X1:POKE V+3,Y1:POKE V
```

```
    ,X:POKE V+1,Y1
3020 FOR I=0 TO 100:NEXT:IF Y1<250
     THEN 3000
3030 POKE 54276,0:VI=VI-1:GOSUB 350
     0:GOTO 400
3097 REM    ********************
3098 REM    * COLLIS. SPRITE 3 *
3099 REM    ********************
3100 X2=X2-8:GOSUB 4600
3110 POKE V+4,X2:POKE V+5,Y2:POKE V
     ,X2:POKE V+1,Y
3120 FOR I=0 TO 100:NEXT:IF X2>5 TH
     EN 3100
3130 POKE 54276,0:VI=VI-1:GOSUB 350
     0:GOTO 400
3197 REM    **********************
3198 REM    * COLLIS. SPRITE 4-5 *
3199 REM    **********************
3200 GOSUB 4700:FOR I=0 TO 64:POKE
     53280,I:NEXT:POKE 54276,0
3210 FOR I=0 TO 1000:NEXT
3220 VI=VI-1:GOSUB 3500:POKE 53280,
     11
3230 L=1:GOTO 400
3497 REM    ******************
3498 REM    * CONTROLLA VITE *
3499 REM    ******************
3500 PRINT"[HOME][10 DOWN]"SPC(33)V
     I
3510 IF VI=0 THEN 4500
3520 RETURN
3597 REM    *******************
3598 REM    * SEGNA N. QUADRO *
3599 REM    *******************
3600 PRINT"[HOME][4 DOWN]"SPC(33)QU
     :RETURN
3697 REM    ***************
3698 REM    * SEGNA PUNTI *
3699 REM    ***************
3700 PRINT"[HOME][16 DOWN]"SPC(32)P
     U:RETURN
3997 REM    ***************
3998 REM    * FINE QUADRO *
3999 REM    ***************
4000 FOR I=0 TO 96:POKE 53280,I:NEX
     T:FOR I=0 TO 1000:NEXT
4010 PRINT"[CLEAR][3 DOWN]":POKE 53
     280,11:POKE 53281,11:POKE V+21
     ,248
4020 PRINTSPC(10)"[NERO]QUADRO COMP
     LETATO !!![CELESTE]"
4030 FOR I=0 TO 3000:NEXT:QU=QU+1:P
     A=0:PRINT"[CLEAR][DOWN]"
4040 GOTO 300
4497 REM    **************
4498 REM    * RICOMINCIA *
4499 REM    **************
4500 POKE 53280,11
4510 PRINT"[HOME][20 DOWN]"SPC(30)"
     [NERO]GAME OVER[CELESTE]"
4520 PRINT"[DOWN]"SPC(33)"[NERO]Y/N
     [CELESTE]"
4530 GET R$:IF R$="N" THEN END
4540 IF R$="Y" THEN 4560
4550 GOTO 4530
4560 QU=1:VI=3:PA=0:PU=0
4570 POKE V+21,240:PRINT"[CLEAR][DO
     WN]":GOTO 300
4597 REM    ************************
4598 REM    * SUONO COLL. SPR. 2-3 *
4599 REM    ************************
4600 S=54272:POKE S+24,15:POKE S+5,
     64:POKE S+6,64
4610 POKE S,49:POKE S+1,28+D:POKE S
     +4,33
4620 D=D+1:RETURN
4697 REM    ************************
4698 REM    * SUONO COLL. SPR. 4-5 *
4699 REM    ************************
4700 S=54272:POKE S+24,15:POKE S+5,
     64:POKE S+6,64
4710 POKE S,12:POKE S+1,1:POKE S+4,
     129:RETURN
4997 REM    *********************
4998 REM    * DATA SPRITE 1 2 3 *
4999 REM    *********************
5000 DATA  0,255,0,0,219,0,0,219,0,
     24,255,24,24,255,24,24,60,24
5010 DATA  24,60,24,31,255,248,31,2
     55,248,0,255,0,0,255,0,0,255,0
5020 DATA  0,255,0,0,255,0,0,255,0,
     0,255,0,0,195,0,0,195,0,0,195
5030 DATA  0,0,195,0,0,195,0
5097 REM    *******************
5098 REM    * DATA SPRITE 3 4 *
5099 REM    *******************
5100 DATA  0,255,0,0,255,0,0,255,0,
     0,255,0,0,255,0,0,60,0,0,60,0
5110 DATA  255,255,255,255,255,255,
     0,255,0,0,255,0,0,255,0,0,255,
     0
5120 DATA  0,255,0,0,255,0,1,255,12
     8,3,255,192,7,0,224,14,0,112
5130 DATA  28,0,56,56,0,28
```

IL DISK DRIVE 1541

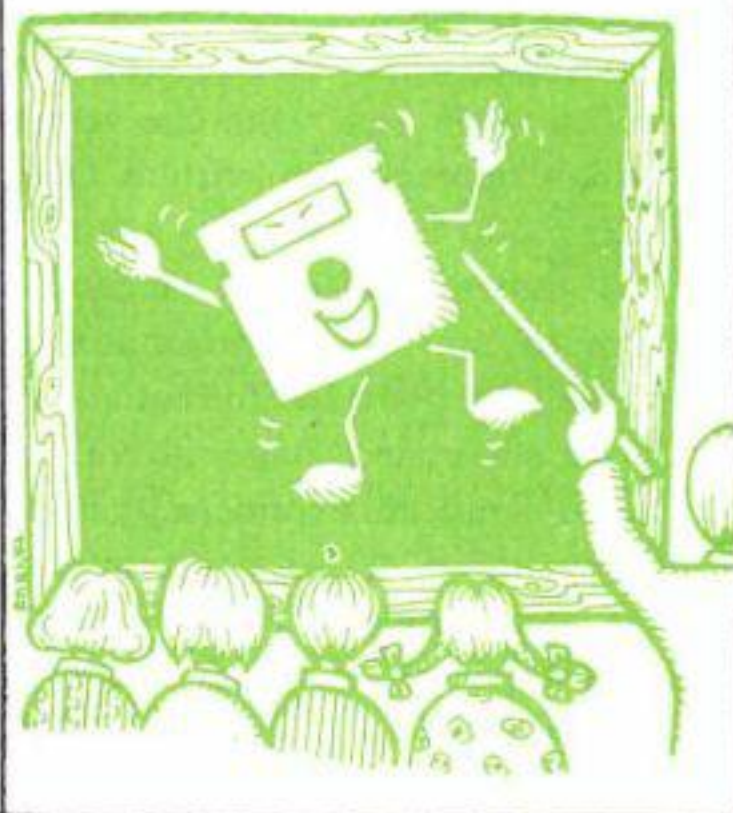
di **Giovanni Verrelli**

Viaggio nel dischetto flessibile. Siete sicuri di sapere tutto sul vostro Drive 1541?

Il passaggio dal registratore a cassette all'unità a dischi è un evento che, prima o poi, si verifica per quasi tutti i possessori del C64 (ed anche per qualcuno del Vic 20).

Nel presente articolo, pertanto, si affronterà l'argomento disk drive cercando di sopperire alle notevoli carenze del relativo manuale d'uso.

Cominciamo subito con alcune notizie hardware. Il 1541 (questo il numero che contraddistingue in casa Commodore la periferica in esame), come sicuramente già saprete, dispone di un proprio microprocessore 6502, di due processori 6522 che sovrintendono all'I/O (Input - Output: Ingresso Uscita dei dati elaborati), 2K di memoria RAM (da usare per piccoli programmi personali o come buffer) e ben 16K di ROM.

Il suo punto di forza è la presenza nella ROM appena citata di un proprio sistema operativo (chiamato DOS dalle prime lettere di "Disk Operating System", "sistema operativo per dischi") in grado di controllare autonomamente e senza alcun intervento da parte dell'unità centrale tutte le operazioni che esso compie.

E' per questo motivo che viene chiamato dispositivo intelligente.

I dischetti usati (chiamati anche floppy disk perchè sono flessibili) per le memorizzazioni corrispondono al formato, comune alla maggior parte dei personal computer, di 5.25" (5 pollici e un quarto, corrispondenti a circa 13 centimetri). Devono essere a singola faccia (cioè con una sola facciata impiegabile per la registrazione, anche se molti, con un piccolo taglio sul lato destro, ne utilizzano anche l'altra) ed a singola densità. Si presentano racchiusi in una busta protettiva, dalla quale non possono e non devono essere estratti, con un largo foro centrale che ne permette la rotazione e che in quelli di buona qualità è rinforzato da un anello di materiale plastico per evitarne la possibile deformazione dovuta al continuo sforzo di trazione.

Sotto è visibile un altro foro, di forma allungata, che permette alla testina di entrare in contatto con la superficie magnetica del disco. Questo è il punto più delicato e non deve essere in nessun caso toccato con le dita o lasciato esposto all'aria o al sole senza la busta di protezione.

Infine sul lato sinistro è presente un incavo rettangolare che, opportunamente coperto con un nastro adesivo, impedisce tutte le operazioni di scrittura, rivelandosi perciò particolarmente utile per evitare alterazioni o cancellazioni involontarie dei dati.

La formattazione

Per consentire l'immagazzinamento dei dati, la prima operazione da compiere sul dischetto è la formattazione, mediante la quale si assegna al disco una certa "forma" che lo rende utilizzabile per gli usi successivi.

Più precisamente, con questa prima operazione si divide la superficie del disco in una serie di aree circolari concentriche chiamate tracce e contraddistinte da un numero che va da 1 a 35. Queste tracce, a loro volta, sono suddivise in un numero variabile di settori (o blocchi) che vengono adoperati materialmente per l'immagazzinamento dei dati.

NUMERO TRACCIA	SETTORI	TOTALE
1 - 17	0 - 20	21
18 - 24	0 - 18	19
25 - 30	0 - 17	18
31 - 35	0 - 16	17
TOTALE SETTORI		683

Figura 1. Distribuzione dei settori per traccia.

Come si può notare dalla tabella di fig 1, una traccia può avere da 17 a 21 settori per un totale di 683 blocchi disponibili per le memorizzazioni anche se, come vedremo tra poco, solo 664 sono liberi per l'utente.

Ogni settore, oltre a poter conservare 256 byte, contiene in una zona non accessibile all'utente alcune informazioni usate per controllare la bontà della registrazione (ID del disco, numero della traccia in cui esso si trova, ecc.).

Gestita in modo diverso rispetto a tutte le altre, la traccia 18, costituita da 19 settori numerati da 0 a 18, è riservata per gli usi della "directory" ed è senza ombra di dubbio la più importante. Per questo motivo i settori liberi per l'utente sono 664: infatti 683-19=664. Grazie alle informazioni contenute in essa è possibile sapere in qualsiasi momento quanti e quali programmi sono registrati sul dischetto, quanti blocchi sono ancora liberi per la registrazione e così di seguito.

Vediamo, come sono conservate tutte queste informazioni.

Il settore 0 di questa traccia (fig 2) contiene la BAM, o Block Availability Map (mappa di occupazione dei blocchi, in italiano), il cui scopo è di registrare, per ogni settore del disco, se esso è libero per la memorizzazione o già occupato. Consiste di 144 byte, corrispondenti ai primi 143 byte del settore, ognuno dei quali con uno specifico significato.

I primi due, numerati 0 e 1, sono usati come concatenamento al settore seguente (indicano qual è il prossimo settore da leggere) e contengono 18 e 1 (traccia=18 e settore=1).

Segue il byte 2, contenente il valore 65, codice ASCII della lettera A, che comunica al drive in che formato è registrato il disco e che viene controllato automaticamente ogni volta che vengono effettuate operazioni di scrittura. Questo byte è molto importante in quanto, se per un motivo qualsiasi se ne cambia il contenuto, non sarà più possibile scrivere su quel disco se non riformattandolo.

Il byte successivo, numerato col 3, non ha alcun significato e contiene il valore 0.

I byte dal 4 al 143 costituiscono la BAM vera e propria e, contengono la mappa dei settori occupati. Questi 140 byte sono suddivisi in 35 gruppi di 4 byte ciascuno, ad ognuno dei quali corrisponde il relativo numero di traccia (al primo gruppo corrisponde la traccia n.1, al gruppo 2 la traccia n.2 e così via fino al n. 35).

I quattro byte che formano ciascun gruppo contengono le diverse informazioni e sono gestiti in questo modo:

- il primo byte contiene il numero dei settori ancora disponibili nella traccia;
- il secondo byte indica la condizione dei primi 8 settori della traccia: a questo scopo vengono usati i singoli bit (si comincia a partire da quello situato più a destra), con la convenzione che se esso è uguale ad 1 il relativo settore è libero, mentre se vale 0 il settore è occupato;
- il terzo byte indica la condizione degli altri 8 settori della traccia a partire dal settore 8 (bit più a destra) al 15 (bit più a sinistra);
- il quarto byte contiene la mappa dei rimanenti settori della traccia.

Giacchè con tre byte è possibile registrare 24 settori (0-23) mentre il numero di settori per traccia è variabile tra 17 e 21, il DOS risolve questo problema ponendo a 0, durante la formattazione, i bit corrispondenti ai settori inesistenti, facendoli così risultare occupati e perciò non liberi per la memorizzazione.

Per esempio, se il secondo byte del primo gruppo contiene il valore 253 (11111101 in binario), allora solo il settore numero 1 della traccia 1 è già stato usato per memorizzare dei dati (e perciò non libero per successive memorizzazioni); se il terzo byte del ventesimo gruppo contiene il valore 124 (01111100 in binario), allora nella traccia n. 20 i settori 10, 11, 12, 13 e 14 sono liberi per ricevere dei dati, mentre quelli con numero 8, 9 e 15 già occupati.

Proseguendo nell'esame della traccia 18 settore 0, i byte dal 144 al 166 sono usati per l'identificazione del dischetto; quelli rimanenti, dal 167 al 255, non vengono utilizzati. Perciò i 16 caratteri del nome del disco ed i 2 dell'ID sono contenuti in questa seconda parte del settore 0. Più precisamente, essi sono memorizzati, carattere per carattere in codice ASCII, nei byte 144-159 e 162-163.

Nel caso in cui il nome del disco avesse

BYTE	CONTENUTO	DESCRIZIONE
0-1	18 - 01	TRACCIA-SETTORE 1 BLOCCO DIR.
2	65	'A' FORMATO DI REGISTRAZIONE
3	00	BYTE NON USATO
4-143	VARIABILE	BAM-MAPPA OCCUPAZIONE SETTORI
144-149	VARIABILE	NOME DEL DISCO
160-161	160-160	BYTE NON USATO
162-163	VARIABILE	ID. DEL DISCO
164	160	BYTE NON USATO
165-166	50 - 65	'2A' VERSIONE DEL DOS
167-255	VARI	BYTE NON UTILIZZATI

Figura 2. Descrizione della traccia 18 settore 0.

un numero di caratteri inferiore a 16, allora il DOS aggiunge automaticamente, durante la formattazione, degli spazi shiftati (valore ASCII 160); mentre, se supera il limite accennato viene opportunamente troncato.

Il nome del disco è inserito per comodità dell'utente e non viene usato dal DOS, al contrario dell'ID (abbreviazione di "Identification Disk", "Identificazione del Disco"), che riveste un'estrema importanza per esso e che si consiglia far variare da disco a disco pena la possibile perdita dei dati.

Come usare il programma "DISPLAY & TS"

Prima di proseguire nell'esame della traccia 18, esaminiamo un programma che ci permette di vedere tutte queste cose.

Ci riferiamo al "DISPLAY T&S", contenuto nel disco (il Test/demo) fornito insieme al drive al momento dell'acquisto, che ci permette di esaminare il contenuto di qualsiasi byte di qualsiasi traccia e settore (ricordiamo che per il Vic 20 questo programma necessita di un'espansione di memoria di almeno 8K).

Dopo averlo caricato in memoria, e dato il RUN, la prima domanda a cui bisogna rispondere è se vogliamo che i risultati vadano su video (screen) o su stampante (printer) premendo la lettera corrispondente all'iniziale del nome (inglese) della nostra scelta.

Fatto questo, il programma chiederà il numero di traccia e settore che vogliamo esaminare, dopodichè il contenuto di questo sarà visualizzato sulla periferica prescelta (nel caso si fosse scelto il video, la pressione del tasto CTRL rallenterà la visualizzazione; premendo il RUN/STOP la si fermerà del tutto con la possibilità, scrivendo CONT e battendo RETURN, di riprenderla in qualsiasi momento).

Il contenuto di ogni settore è visualizzato in notazione esadecimale, un metodo molto comune per la rappresentazione dei numeri binari. Per la conversione nel formato più familiare di numeri decimali, basterà moltiplicare la prima cifra del valore per il numero 16 ed aggiungerne la seconda, ricordando che ai caratteri A, B, C, D, E ed F corrispondono, rispettivamente, i valori 10, 11, 12, 13, 14 e 15. Perciò, al valore esadecimale A6 corrisponde il numero decimale 166 (10*16+6), mentre al valore 12 corrisponde il numero 18 (1*16+2).

Alla destra di ogni linea, inoltre, è visualizzato il carattere ASCII corrispondente ai diversi valori presenti.

Infine, per uscire dal programma è sufficiente rispondere alla richiesta del numero di traccia con un valore esterno all'intervallo 1-35.

Riprendendo il discorso iniziale, i rimanenti settori della traccia 18 sono utilizzati per la directory (contengono i nomi dei file registrati e indicano dove questi sono allocati).

Poichè in ogni settore è possibile registrare l'indice di 8 file (fig 3), un disco può contenere solo (!) 144 file diversi. Infatti 8 file * 18 settori rimanenti nella traccia 18 = 144.

Come per il settore 0, i primi due byte del settore "1" puntano (=indicano) al prossimo blocco da leggere: questa pratica del puntatore è usata per tutti i settori in quanto le varie registrazioni non avvengono sequenzialmente, perciò si ha bisogno di sapere ogni volta in quale blocco la registrazione continua. L'ultimo settore utilizzato per la memorizzazione è indicato dal fatto che i primi due byte puntano alla traccia 0, settore 255 (o altro valore), che in realtà non esiste.

Dopo questi primi due byte abbiamo i 30 byte che contengono tutte le informazioni sul primo file registrato (fig 3,1).

Di questi:

Il byte 0 indica il tipo di file:
129 ($81) per sequenziale (SEQ)
130 ($82) per programma (PRG)
131 ($83) per user (USR)
132 ($84) per relativo (REL)
0 ($00) per file cancellato (DEL)
valori intermedi tra 128 e 1 per file chiusi non correttamente che il DOS ignorerà;

- i byte 1 e 2 puntano alla traccia e settore dove è stato memorizzato il primo blocco del file;
- i byte dal 3 al 18 contengono il codice ASCII dei caratteri del nome del file (anche in questo caso se il nome è più corto di 16 caratteri il DOS aggiungerà automaticamente degli spazi shiftati, codice ASCII 160);
- i byte dal 19 al 21 sono usati solo per i file relativi e contengono l'indirizzo di traccia e settore del primo SIDE SECTOR (un indice interno usato dal file, byte 19 e 20) e la lunghezza di ogni record (21);
- i byte dal 22 al 25 non sono utilizzati;
- i byte 26 e 27 contengono il numero di traccia e settore del file che è stato memorizzato dopo il comando OPEN con un nome già presente in directory (questo è quanto dice il manuale d'uso, mentre personalmente ho visto che contengono sempre 0)
- i byte 28 e 29 indicano il numero di blocchi occupati dal file nel formato byte basso-byte alto.

Al termine di questi primi 30 byte ci sono 2 byte contenenti il valore 0 ed usati come caratteri separatori; quindi si hanno altri 30 byte usati per le informazioni della seconda registrazione seguiti da due byte a zero, e così via fino all'indice dell'ottavo file che concluderà il settore 1. Si ricordi che ogni settore può contenere al massimo l'indice di 8 file.

Per gli eventuali altri settori usati per la directory, presenti se i file memorizzati sono più di 8, valgono le stesse spiegazioni viste per il settore 1.

L'ultima informazione (non riportata nemmeno sul manuale d'uso) è che, solo per i file programmi, i byte 2 e 3 del primo settore dove comincia la loro memorizzazione contengono la locazione iniziale, espressa come al solito nel formato byte basso-byte alto, dove risiedeva il programma al momento del salvataggio.

Vediamo ora in pratica come tutte le informazioni appena viste sono realmente utilizzate.

Impariamo a leggere il contenuto di un disco

Dato che in questa fase sarà largamente impiegato il DISPLAY T&S, cerchiamo di renderlo ancor più efficiente aggiungendo la riga:

453 WAIT 198,1,1

per mezzo della quale ogni volta che si desidera fermare la visualizzazione del contenuto di un settore basterà premere un tasto (diverso dallo SHIFT o dallo STOP), con la possibilità, ripremendolo un'altra volta, di continuare il lavoro iniziale.

Per prima cosa, formattiamo un disco nuovo eseguendo il comando:

OPEN15,8,15,"N0:PIPPO,00"
:CLOSE15

per fare in modo che tutto ciò che sarà detto tra breve sia identico, per le prove che farete a casa vostra, per qualsiasi disco.

Fatto questo, caricate il DISPLAY T&S e andate a leggere il contenuto della traccia 18 settore 0 che, come detto precedentemente, contiene la BAM, la mappa di occupazione dei settori.

I primi due byte, come facilmente riscontrabile anche dalla figura 4, contengono i valori $12 (18 in decimale) e $01 (1 in decimale), indicanti al DOS che il prossimo settore da leggere è il n. 1 della traccia 18. Seguono il byte indicante il formato dell'unità ($41, 65 in decimale, codice ASCII della lettera A) ed il byte non usato che contiene il valore 0.

Nella BAM vera e propria (byte dal 4 al 143) possiamo vedere che effettivamente tutti i settori del disco, tranne i numeri 0 e 1 della traccia 18, sono liberi per le registrazioni (infatti il disco è stato appena formattato) e che i settori non esistenti sono codificati come già occupati (vedasi a tal proposito il byte n. 7, contenente il valore $1F, "31" in decimale e 00011111 in binario, indicante che i settori n. 16, 17, 18, 19 e 20 della traccia 1 sono liberi per le memorizzazioni, mentre quelli n. 21, 22 e 23, che in realtà non esistono, già "occupati").

Sempre in questo settore possiamo vedere che effettivamente il nome del disco (PIPPO) è memorizzato con l'aggiunta di spazi shiftati ($A0, 160 in decimale) nei byte dal 144 al 159 ($90 - $9F), che i due caratteri della ID sono presenti nei byte 162 e 163 ($A2 e $a3) e che i byte dal 167 al 255 ($A7 - $ff) non sono utilizzati (infatti quasi tutti contengono il valore 0).

Questa è la situazione presente quando il disco è "vuoto", cioè appena formattato.

Vediamo ora cosa succede nel momento in cui si registra il primo programma. A tal proposito, dopo aver effettuato il NEW per cancellare gli eventuali programmi presenti in memoria, digitiamo la linea:

10 REM PROGRAMMA DI PROVA

e la salviamo su disco col comando:

SAVE "PROVA",8

Fatto questo, andiamo a leggere, dopo aver ricaricato DISPLAY T&S, il contenuto della traccia 18 settore 1 che, ricordo, contiene tutte le informazioni dei primi 8 file registrati, tra cui ora figura "PROVA".

Come si può constatare anche dalla Fig 5, i primi due byte del settore contengono i valori 0 e 255 ($00 e $FF) indicanti che questo è l'ultimo settore (e perciò anche l'unico) utilizzato per la definizione dei file registrati.

Subito dopo abbiamo il valore 130 ($82) indicante che siamo in presenza di un file programma, seguito dai valori 17 ($11) e 0 ($00) che ci informano che il primo settore utilizzato per la registrazione del programma è il n. 0 della traccia 17.

A seguire troviamo il nome del programma (PROVA nel nostro caso) completato da tanti spazi shiftati quanti ne occorrono per raggiungere i 16 caratteri (11, dato che il nome PROVA è di 5 caratteri) unitamente ad altri byte, dal 21 ($15) al 29 ($1D), contenenti il valore 0, in quanto non utilizzati in questo caso.

Gli ultimi due byte (30 e 31, $1E e $1F) informano che il programma occupa in tutto 1 blocco (1+0*256).

Ritornando alla traccia 18 settore 0 dopo la registrazione del programma, vedremo che nella BAM gli unici valori variati sono quelli contenuti nei byte 68

BYTE	DESCRIZIONE
0 - 1	PROSSIMO BLOCCO DA LEGGERE
2 - 31	ENTRATA 1' FILE DIRECTORY
34 - 63	ENTRATA 2' FILE DIRECTORY
66 - 95	ENTRATA 3' FILE DIRECTORY
98 -127	ENTRATA 4' FILE DIRECTORY
130-159	ENTRATA 5' FILE DIRECTORY
162-191	ENTRATA 6' FILE DIRECTORY
194-223	ENTRATA 7' FILE DIRECTORY
226-255	ENTRATA 8' FILE DIRECTORY

Figura 3. Descrizione della traccia 18 settore 1.

($44) e 69 ($45), relativi alla mappa di occupazione dei settori della traccia 17. Infatti i valori ora presenti sono $14 (20 in decimale), indicanti che nella traccia 17 sono liberi solamente 20 settori e $fe, 254 in decimale e 11111110 in binario, indicante che il settore 0 di questa traccia è già occupato.

Volendo ora esaminare come realmente è stato registrato il programma PROVA, dobbiamo semplicemente visualizzare il contenuto della traccia 17 settore 0 che, come visto prima, è il primo settore utilizzato per la memorizzazione del file in oggetto.

I primi due byte di questo blocco puntano al settore 30 ($1E) della traccia 0 ($00) e, dato che non esiste alcuna traccia 0, comprendiamo subito che solo in questo settore è stato memorizzato il nostro mini programma.

Subito dopo abbiamo i valori 1 e 8 ($01 e $08) costituenti il byte basso ed il byte alto della locazione iniziale dove risiedeva il programma al momento del salvataggio (1+8*256=2049, il normale inizio del programma Basic, nel Commodore 64).

Dal byte n. 4 in poi troviamo esattamente il programma così come codificato in memoria centrale (byte di link, tokens, ecc.), concluso della presenza di tre byte contenenti il valore 0 posti uno di seguito all'altro.

Dopo l'esame di ciò che accade con la registrazione dei programmi, vediamo ora cosa succede quando impartiamo il comando:

OPEN15,8,15,"S0:PROVA"
:CLOSE15

mediante il quale si ottiene la cancellazione del programma PROVA precedentemente registrato.

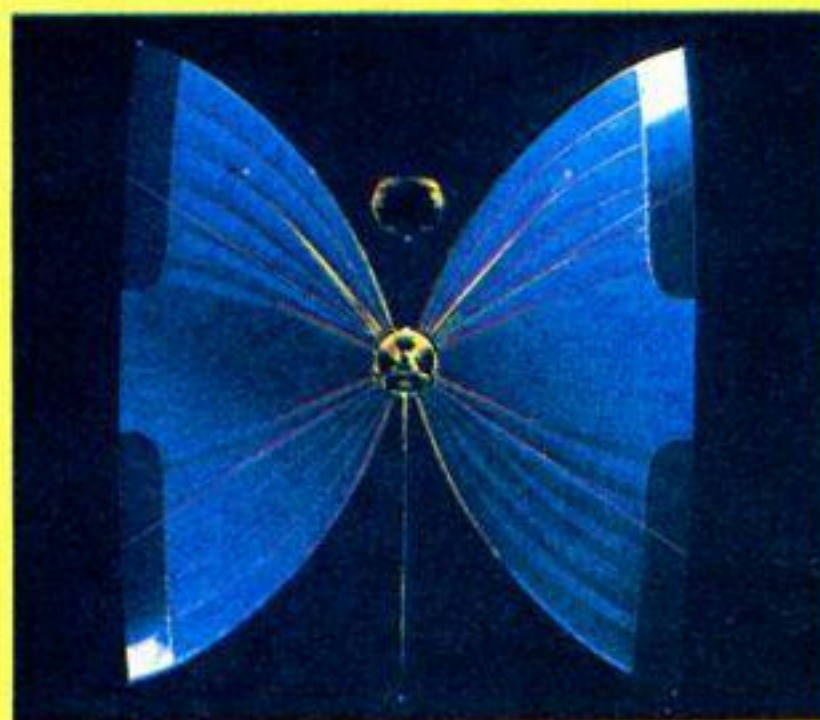
Visualizzando il contenuto della traccia 18 settore 1 possiamo notare che l'unica differenza intervenuta dopo l'esecuzione dello SCRATCH è la comparsa del valore 0 nel byte $04 (4 in decimale) in luogo del valore 130 ($82) originario. In questo modo il DOS comprende di essere in presenza di un programma cancellato (al codice 0, infatti, corrisponde il significato di file cancellato) e di ignorare tutte le altre informazioni relative a quel determinato file.

BYTE	DESCRIZIONE
0	TIPO DI FILE
	0 =FILE CANCELLATO (DEL)
	129=FILE SEQUENZIALE (SEQ)
	130=FILE PROGRAMMA (PRG)
	131=FILE USER (USR)
	132=FILE RELATIVO (REL)
1-2	TRACCIA-SET. 1' BLOCCO DATI
3-18	NOME DEL FILE
19-20	PER I FILE RELATIVI TRACCIA
	E SET. DEL PRIMO SIDE SECTOR
21	PER I FILE RELATIVI LUNGHEZ-
	ZA DEI RECORD
22-25	BYTES NON UTILIZZATI
26-26	TRACCIA E SETTORE DEL FILE
	RIMPIAZZATO (@0:FILE)
28-29	BLOCCHI DEL FILE BYTE BASSO-
	BYTE ALTO

Figura 4. Contenuto della traccia 18 settore 0 dopo la formattazione.

Inoltre, nella BAM possiamo vedere che i valori ora presenti indicano che anche il blocco usato precedentemente per memorizzare il nostro mini programma (traccia 17 settore 0) è ora libero per la sovrascrittura.

Impiegando la stessa tecnica usata in questi esempi è possibile estendere lo studio appena fatto a tutte le possibili situazioni che possono venire a crearsi (programmi occupanti più di 1 blocco, file sequenziali e relativi, ecc.) e, dato che per farlo in questa sede sarebbero necessarie molte più pagine di quelle possibili con il pericolo di rendere il tutto molto noioso, consiglio gli interessati di effettuare personalmente questo "studio supplementare", utilizzando magari i momenti vuoti (!) della propria giornata.

Prima di concludere vi anticipo che prossimamente esamineremo alcuni comandi che permettono di colloquiare direttamente con i singoli byte di qualsiasi blocco unitamente ad alcuni programmi di utilità coi quali vedremo come poter usare in pratica tutte le nozioni finora apprese.

Visione frontale di quello che appare ai protagonisti del viaggio nel computer nel film Tron. Ogni inquadratura richiedeva da 5 a 75 milioni di calcoli. Ogni minuto del film, diretto da Steven Lisberger, è composto da 1240 inquadrature.

VIC 20

INDOVINELLO

Questo simpatico gioco metterà alla prova le vostre capacità di intuizione e di ragionamento.

Possono usufruirne tutti coloro che possiedono un computer Vic 20, anche se privo di espansioni di memoria.

Il calcolatore vi chiederà, inizialmente, quanti errori può tollerare prima di porre termine alla prosecuzione del gioco.

Il vostro obiettivo è riuscire ad individuare una parola scelta dal computer tra quelle che ha a disposizione inizialmente.

Le regole vi pongono un'unica limitazione, non potete fornire, in fase di INPUT, più di una lettera alla volta.

Le lettere gia prescelte compariranno nella zona inferiore del vostro video.

Buon gioco e aguzzate il vostro ingegno!

```
3130 PRINT"[HOME]   SEI IMBRANATO !
     ":PRINT"                    "
3140 GOTO 4010
4000 IF W$<>L$ THEN 3000
4010 PRINT"[HOME][DOWN]"; TAB(IND);
4020 FOR  I=1  TO  LN
4030 :PRINT " ▄▄[3 LEFT][DOWN][RVS]
     ▌";MID$(W$,I,1);"[RVOFF]▌[3 LE
     FT][DOWN]"[RVS]▄[RVOFF]"[2 UP]
     ";
4040 NEXTI
5000 PRINT"[HOME][14 DOWN]ANCORA? "
     ;
5010 GOSUB 60000:IN$=LEFT$(IN$,1)
5020 IF IN$="S" OR IN$="" THEN 1500
5030 END
9997 GOTO 1500
9998 PRINT:PRINT
9999 END
30000 REM -*-TABELLINA DELLE PAROLE
      -*-
30010 DATA VECCHIO,OGGETTO,LAVORO,A
      TIGIANO,GRAZIOSO,NATALE,ORGAN
      ZZAZAZIONE,TRE
30020 DATA TELEFONO,MODICO,MINIATURE
      ,PENSIERO,CERAMICA,BAMBOLE,MAT
      RIMONIO,VASO
30030 DATA PIATTO,ANFORA,GRAFICO,ACQ
      UARIO,ALBUM,UNO,DUE,TREDICI,RE
      PUBBLICA,VISO
30040 DATA COLLEZIONE,FRANCOBOLLO,FI
      LATELIA,MONETA,CAMPAGNA,CUCCIO
      LATA,GATTO
30050 DATA GABBIA,GIOCHI,AUTOMODELLO
      ,ALIANTE,PLASTICA,TECNIGRAFO,A
      RTE,BELLO,ELMO
30060 DATA PERGAMENA,FIORI,FIORISTA,
      ELETTRONICA,PROFESSIONALE,COMP
      UTER,OTTIMO
30070 DATA CRISTALLO,CALORIFERO,GHIS
      A,STABILE,PREZIOSI,DIAMANTE,OR
      O,VALORE,BIRO
30080 DATA PIZZA,PIZZERIA,RISTORANTE
      ,TRANSISTORE,ANTENNA,RADIOAMAT
      ORE,VISTA
30090 DATA VITTORIA,NATALE,FESTA,DOC
      CIA,PORCELLANA,IDRAULICO,PORTA
      ,VERNICIATURA
30100 DATA FLUORESCENTE,PIASTRELLA,P
      ATENTE,STENOGRAFIA,SPORT,SPORT
      IVO,CORSA
30110 DATA TESTO,COLORE,TELEVISIONE,
      FINANZIAMENTO,AUTOMAZIONE,UNIV
      ERSITA,ANNO
30120 DATA PAGINA,TERMINALE,RICERCA,
      DISTRIBUZIONE,CLIENTE,SYSTEM,C
      ONSIGLIO
30130 DATA VISITA,VOLUME,LIBRO,TESTO
      ,RIVISTA,GLORIANO,DATI,ELABORA
      TORE,RUM,GIN
30140 DATA GRAPPA,PREZZO,APERTO,APER
      TURA,PORTO,TRASMISSIONE,VINO,B
      OTTIGLIA,TAPPO
30150 DATA FINE,FINALE,TERMINE,USCIT
      A,CHISO,CHIUSURA,STOP,ARRIVO,O
      CCHIO,GIORNO
39000 DATA ""
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20
      )
60010 GET Z$:IF  Z$<>"" THEN 60070
60020 IF ZT<=TI THEN PRINTMID$(" ▒",
      ZC,1);"[LEFT]";:ZC=3-ZC:ZT=TI+
      15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AN
      D 127)<32 THEN PRINT" [LEFT]";
      :GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (
      Z AND 127)<91 THEN Z$=CHR$((Z+
      128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINTZ$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):P
      RINTCR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT
      $(IN$,ZL-1):PRINT"[LEFT]";:GOT
      O 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>
      1)):FOR Z=2 TO ZL:PRINTZ$;:NEX
      TZ:GOTO 60000
60140 GOTO 60010
60200 IF PN=0 THEN GOTO 60259
60210 PRINT"[DOWN]PREMI [RVS]RETURN[
      RVOFF] PER ";
60220 IF PN=LP THEN PRINT"INIZIARE "
      ;:GOTO 60240
60230 PRINT"CONTINUE:";
60240 GET T$:IF T$="" THEN 60240
60250 IF PN=LP THEN POKE 59468,PK:CL
      R :PRINT"[CLEAR]";:GOTO 100
60260 PN=PN+1:PRINT"[CLEAR]";PG$;" -
      *H*- ╮2╯| -*H*-  (♠─╰) ─╮─.197
      9[DOWN]"
60270 RETURN
```

```
10 CLR :POKE 36879,12
20 PRINT"[CLEAR][HOME]"
90 PG$="PAROLE"
100 CR$=CHR$(13)
900 PRINT
1000 WC=0
1010 READ T$:IF T$<>"" THEN WC=WC+1
     :GOTO 1010
1015 PRINT"[3 UP]MINISYSTEMS
        EDITORIALE
1016 PRINT"[2 DOWN]       **********
1017 PRINT"       * PAROLA *"
1018 PRINT"       **********"
1020 PRINT"[DOWN][BIANCO] QUANTI ER
     RORI DEVO    AMMETTERE AL MAX
     ?":PRINT "[2 DOWN]A TE ? ";
1030 GOSUB 60000:IF IN$="" THEN LIM
     =1E2:GOTO 1500
1040 FOR I=1 TO LEN(IN$):T$=MID$(IN
     $,I,1):IF T$>="0" AND T$<="9"
     THEN 1060
1050 PRINT"NON LETTERE  PREGO":GOTO
      1020:
1060 NEXTI
1070 LIM=VAL(IN$)
1080 IF LIM<1 THEN PRINT"NESSUNO PO
     TREBBE    FARE DI MEGLIO !":GO
     TO 1020
1500 T=INT(RND(1)*WC)+1:RESTORE :FO
     R I=1  TO T:READ W$:NEXTI
1510 LN=LEN(W$):LC=LN:BL$=MID$(" ",
     2-(LN  AND  1))
1520 WR=0:PRINT
1525 IF  LN>7 THEN 1500
2000 PRINT"[CLEAR] LA ";BL$;"PAROLA
     ":
2020 FOR I=1 TO LN
2030 PRINT" ┌─┐[3 LEFT][DOWN]|?|[3 L
     EFT][DOWN]└─┘[2 UP]";
2040 NEXTI:PRINT
2060 PRINT"[7 DOWN]  **  GIA'SCELTE
      **"
3000 PRINT"[HOME][5 DOWN]INDOVINA:
     ";
3005 PRINTLEFT$("
                          ",LEN(L$
     )+1)
3006 PRINT"[UP]INDOVINA: ";:GOSUB 6
     0000:L$=IN$
3010 IF LEN(L$)>1 THEN 4000
3012 IF  IN$="" THEN 4010
3015 IF L$<"A" OR L$>"Z" THEN 3000
3020 M=0:PRINT"[HOME][DOWN]";
3030 FOR I=1 TO LN
3040 IF MID$(W$,I,1)<>L$ THEN PRINT
     "[3 RIGHT]";:GOTO 3060
3050 PRINT" ▄▄ [3 LEFT][DOWN][RVS]▌"
     ;L$;"[RVOFF]▌[3 LEFT][DOWN]▀[R
     VS]▄[RVOFF]▀[2 UP]";:M=M+1:LC=
     LC-1
3060 NEXTI:PRINT
3080 IF M>0 THEN 3100
3090 WR=WR+1
3100 PRINT"[HOME][8 DOWN][RVS]"; TA
     B(ASC(L$)+1);L$;"[RVOFF]"
3110 IF LC=0 THEN 5000
3120 IF WR<LIM THEN 3000
```

QUALSIASI COMPUTER

ALFABETO PAZZO

di **Diego e Federico Canetta**

Il gioco che presentiamo è molto semplice dal punto di vista della programmazione, per cui anche i lettori meno esperti potranno facilmente capirne la struttura.

Sono presentati tre listati separati, adatti per il C64, il C16 e il VIC 20 inespanso.

Spieghiamo innanzitutto come si svolge il gioco. Dopo aver dato il RUN, apparirà una domanda relativa alla velocità del gioco: se viene battuto direttamente il tasto RETURN, viene automaticamente scelta una velocità media. A questo punto apparirà sullo schermo una griglia con tutte le lettere dell'alfabeto. Alcune lettere verranno successivamente illuminate e voi dovrete scrivere la parola che avrete individuato. Se la parola è giusta, si illumineranno le lettere per una successiva parola, altrimenti verrà segnalato l'errore insieme alla parola giusta.

Passiamo ora alla descrizione dei listati.

La riga 100 definisce i colori di bordo e schermo.

Le righe 110 e 120 servono per caricare nelle variabili A$ le parole contenute nelle righe DATA da 470 a 510.

Se volete cambiare le parole, potete farlo, ma mantenendo sempre lo stesso numero di lettere delle parole sostituite. Potete anche aggiungere nuove linee DATA con altre parole, ricordando di mettere uno 0 nell'ultima riga (vedere la riga 510) e cambiando il ciclo FOR presente nella riga 120; inoltre occorre cambiare il dimensionamento della variabile A$ nella riga 110. Ad esempio, se vengono aggiunte 2 righe, occorrerà scrivere FORT=1TO7 e DIMA$ (7,10).

La riga 130 legge la velocità voluta, mentre dalla riga 140 viene scelta (casualmente) la parola da scrivere.

A questo punto il controllo del programma passa alla riga 270 ed alle seguenti, che provvedono a stampare la griglia con le lettere sullo schermo.

La riga 370 restituisce il controllo del programma alla riga 150, che provvede, insieme a quelle che seguono, ad illuminare in sequenza le lettere.

Il resto del programma è facilmente comprensibile.

```
 50 REM  ALFABETO PAZZO
 60 REM   SOFTWARE DI
 70 REM  FEDERICO CANETTA
 80 REM  PER COMMODORE 64
 90 :
100 PRINT"[CLEAR][BLEU]":POKE 5328
    0,0:POKE 53281,0
110 DIM A$(5,10)
120 FOR T=1 TO 5:FOR T1=1 TO 8:REA
    D A$:A$(T,T1)=A$:NEXT:NEXT
130 PRINT"[CLEAR][17 RIGHT]30":INP
    UT "[HOME]VELOCITA'(1-60)";VE:
    IF VE<1 OR VE>60 THEN 130
140 VE=100-VE:FOR LI=0 TO 9:FOR T1
    =1 TO 8:T=INT(RND(0)*4)+1:A=LE
    N(A$(T,T1)):GOSUB 270
150 PRINT"[HOME][AZZUR][7 RIGHT]AD
    ESSO SCRIVO LA PAROLA":FOR F=0
     TO 1000:NEXT
160 FOR P=1 TO A:B=ASC(MID$(A$(T,T
    1),P,1)):IF B<73 THEN D=55467+
    (B-65)*2
170 IF B>72 AND B<81 THEN D=55547+
    (B-73)*2
180 IF B>80 AND B<89 THEN D=55627+
    (B-81)*2
190 IF B>88 THEN D=55707+(B-89)*2
200 POKE D,1:GOSUB 460:FOR F=0 TO
    VE:NEXT:POKE D,6:FOR F=0 TO VE
    :NEXT:NEXT
210 PRINT"[HOME]
                     ":PRINT"[17 DOWN]
    "
220 INPUT "CHE PAROLA ERA";B$
230 IF B$<>A$(T,T1) THEN 380
240 PRINT"[ROSSO]OK[AZZUR]":FOR F=
    0 TO 300:NEXT
250 NEXT:FOR F=0 TO 1000:NEXT:NEXT
260 END
270 PRINT"[CLEAR][BLEU][3 DOWN][10
     RIGHT]┌─┬─┬─┬─┬─┬─┬─┬─┐"
280 PRINT"[10 RIGHT]|A|B|C|D|E|F|G
    |H|"
290 PRINT"[10 RIGHT]├─┼─┼─┼─┼─┼─┼─
    ┼─┤"
300 PRINT"[10 RIGHT]|I|J|K|L|M|N|O
    |P|"
310 PRINT"[10 RIGHT]├─┼─┼─┼─┼─┼─┼─
    ┼─┤"
320 PRINT"[10 RIGHT]|Q|R|S|T|U|V|W
    |X|"
330 PRINT"[10 RIGHT]├─┼─┼─┴─┴─┴─┴─
    ┴─┤"
340 PRINT"        [3 RIGHT]|Y|Z|N"
    LET.="A:PRINT"[UP][26 RIGHT]|"
350 PRINT"        [3 RIGHT]└─┴─┴──
    ──────────┘"
360 PRINT"[11 DOWN][GIALLO]ERRORI
    = "ER
370 RETURN
380 PRINT"[UP]MI DISPIACE LA PAROL
    A ERA ";A$(T,T1):ER=ER+1:IF ER
    =3 THEN ER=0:GOTO 400
390 FOR F=0 TO 1000:NEXT:GOTO 250
400 I=I+1:IF I=6 THEN I=1
410 FOR F=0 TO 1000:NEXT:PRINT"[CL
    EAR][11 DOWN][6 RIGHT]FINE DEL
     GIOCO"
420 FOR F=0 TO 1000:NEXT:PRINT"[CL
    EAR][3 DOWN][5 RIGHT]VUOI GIOC
    ARE ANCORA ?"
430 GET A$:IF A$<>"S" AND A$<>"N"
    THEN 430
440 IF A$="N" THEN PRINT"[8 DOWN][
    3 RIGHT][CLEAR]CIAO":END
450 GOTO 130
460 S=54272:POKE S+24,15:POKE S+1,
    110:POKE S+5,9
470 POKE S+6,3:POKE S+4,17:POKE S+
    4,16:RETURN
480 DATA APE,CLUB,EDERA,ATTICO,BAN
    DITO,GIACINTI,BATUFFOLO,CASSAP
    ANCA
490 DATA AGO,MODA,FIBRE,CHIAVE,CHI
    AZZE,SPREMUTA,CAPPUCCIO,SOPRAN
    NOME
500 DATA SUD,NOTA,GATTI,GABBIA,RAF
    FICA,BRICCONE,INDUSTRIA,MICROS
    COPI
510 DATA TRE,SETE,IGLOO,SLALOM,PRI
    MATO,DECADUTO,SPERGIURO
520 DATA NOI,CIAO,TACCO,GELARE,GAT
    TINO,GRAZIOSO,ERMETISMO,IMPORT
    ANTE,0
```

```
 90 REM ALFABETO PAZZO
 95 REM VERSIONE C-16
 97 REM   & PLUS-4
100 PRINT"":COLOR0,1:COLOR4,1
110 DIMA$(5,10)
120 FORT=1TO5:FORT1=1TO8:READA$:A$(T,T1)=A$:NEXT:NEXT
130 PRINT"30":INPUT"VELOCITA'(1-60)";VE:IFVE<1ORVE>60THEN130
140 VE=100-VE:FORT2=0TO5:FORT1=1TO8:T=INT(RND(0)*4)+1:A=LEN(A$(T,T1)):GOSUB270
150 PRINT"ADESSO SCRIVO LA PAROLA":FORF=0TO1000:NEXT
160 FORP=1TOA:B=ASC(MID$(A$(T,T1),P,1)):IFB<73THEND=2219+(B-65)*2
170 IFB>72ANDB<81THEND=2299+(B-73)*2
180 IFB>80ANDB<89THEND=2379+(B-81)*2
190 IFB>88THEND=2459+(B-89)*2
200 POKED,1:GOSUB460:FORF=0TOVE:NEXT:POKED,70:FORF=0TOVE:NEXT:NEXT
210 PRINT"                                    ":PRINT""
220 INPUT"CHE PAROLA ERA";B$
230 IFB$<>A$(T,T1)THEN380
240 PRINT"OK":FORF=0TO300:NEXT
250 NEXT:FORF=0TO1000:NEXT:NEXT
260 END
270 PRINT"┌─┬─┬─┬─┬─┬─┬─┬─┐"
280 PRINT"|A |B |C |D |E |F |G |H |"
290 PRINT"├─┼─┼─┼─┼─┼─┼─┼─┤"
300 PRINT"|I |J |K |L |M |N |O |P |"
310 PRINT"├─┼─┼─┼─┼─┼─┼─┼─┤"
320 PRINT"|Q |R |S |T |U |V |W |X |"
330 PRINT"├─┼─┴─┴─┴─┴─┴─┴─┤"
340 PRINT"          |Y |Z |N LET.="A:PRINT"|"
350 PRINT"          └─┴─┴───────────┘"
360 PRINT"ERRORI = "R
370 RETURN
380 PRINT"MI DISPIACE LA PAROLA ERA ";A$(T,T1):R=R+1:IFR=3THENR=0:GOTO400
390 FORF=0TO1000:NEXT:GOTO250
400 I=I+1:IFI=6THENI=1
410 FORF=0TO1000:NEXT:PRINT"FINE DEL GIOCO"
420 FORF=0TO1000:NEXT:PRINT"VUOI GIOCARE ANCORA ?"
430 GETA$:IFA$<>"S"ANDA$<>"N"THEN430
440 IFA$="N"THENPRINT"CIAO":END
450 GOTO130
460 VOL8:SOUND1,900,5:RETURN
470 DATAAPE,CLUB,EDERA,ATTICO,BANDITO,GIACINTI,BATUFFOLO,CASSAPANCA
480 DATAAGO,MODA,FIBRE,CHIAVE,CHIAZZE,SPREMUTA,CAPPUCCIO,SOPRANNOME
490 DATASUD,NOTA,GATTI,GABBIA,RAFFICA,BRICCONE,INDUSTRIA,MICROSCOPI
500 DATATRE,SETE,IGLOO,SLALOM,PRIMATO,DECADUTO,SPERGIURO
510 DATANOI,CIAO,TACCO,GELARE,GATTINO,GRAZIOSO,ERMETISMO,IMPORTANTE,0
```

```
60 REM  ALFABETO PAZZO
70 REM  SOFTWARE  DI
80 REM  FEDERICO CANETTA
90 REM  VERS. VIC 20 INESPANSO
95 :
100 PRINT"[CLEAR]":POKE 36879,27
110 DIM A$(5,10):POKE 36874,200
120 FOR T=1 TO 5:FOR T1=1 TO 8:REA
    D A$:A$(T,T1)=A$:NEXT:NEXT
130 PRINT"[CLEAR][17 RIGHT]30":INP
    UT "[HOME]VELOCITA'(1-60)";VE:
    IF VE<1 OR VE>60 THEN 130
140 VE=100-VE:FOR LI=0 TO 9:FOR T1
    =1 TO 8:T=INT(RND(0)*4)+1
144 A=LEN(A$(T,T1)): GOSUB 270
150 PRINT"[HOME]ORA SCRIVO LA PARO
    LA":FOR F=0 TO 1000:NEXT
160 FOR P=1 TO A:B=ASC(MID$(A$(T,T
    1),P,1)):IF B<73 THEN D=38489+
    (B-65)*2
170 IF B>72 AND B<81 THEN D=38511+
    22+(B-73)*2
180 IF B>80 AND B<89 THEN D=38533+
    44+(B-81)*2
190 IF B>88 THEN D=38621+(B-89)*2
200 POKE D,1:GOSUB 460:FOR F=0 TO
    VE:NEXT:POKE 36878,0:POKE D,6:
    FOR F=0 TO VE:NEXT:NEXT
210 PRINT"[HOME]
       ":PRINT"[10 DOWN]"
220 INPUT "CHE PAROLA ERA";B$
230 IF B$<>A$(T,T1) THEN 380
240 PRINT"OK":FOR F=0 TO 300:NEXT
250 NEXT:FOR F=0 TO 1000:NEXT:NEXT
260 END
270 PRINT"[CLEAR][3 DOWN]┌┬┬┬┬┬┬┬
    ┬┐"
280 PRINT" |A |B |C |D |E |F |G |H |"
290 PRINT"├┼┼┼┼┼┼┼┼┤"
300 PRINT" |I |J |K |L |M |N |O |P |"
310 PRINT"├┼┼┼┼┼┼┼┼┤"
320 PRINT" |Q |R |S |T |U |V |W |X |"
330 PRINT"├┼┴┴┴┴┴┴┴┤"
340 PRINT" |Y |Z |N" LET.="A:PRINT"[U
    P][16 RIGHT]|"
350 PRINT"└┴┴──────┘"
360 PRINT"[5 DOWN]ERRORI = "ER
370 RETURN
380 PRINT"[UP]MI DISPIACE LA PAROL
    A ERA ";A$(T,T1):ER=ER+1:IF ER
    =3 THEN ER=0:GOTO 400
390 FOR F=0 TO 1000:NEXT:GOTO 250
400 I=I+1:IF I=6 THEN I=1
410 FOR F=0 TO 1000:NEXT:PRINT"[CL
    EAR][11 DOWN][6 RIGHT]FINE DEL
     GIOCO"
420 FOR F=0 TO 1000:NEXT:PRINT"[CL
    EAR][3 DOWN]GIOCHI ANCORA (S/N
    )?"
430 GET A$:IF A$<>"S" AND A$<>"N"
    THEN 430
440 IF A$="N" THEN PRINT"[8 DOWN][
    3 RIGHT][CLEAR]CIAO":END
450 GOTO 130
460 POKE 36878,15
470 RETURN
480 DATA APE,CLUB,EDERA,ATTICO,BAN
    DITO,GIACINTI,BATUFFOLO,CASSAP
    ANCA
490 DATA AGO,MODA,FIBRE,CHIAVE,CHI
    AZZE,SPREMUTA,CAPPUCCIO,SOPRAN
    NOME
500 DATA SUD,NOTA,GATTI,GABBIA,RAF
    FICA,BRICCONE,INDUSTRIA,MICROS
    COPI
510 DATA TRE,SETE,IGLOO,SLALOM,PRI
    MATO,DECADUTO,SPERGIURO
520 DATA NOI,CIAO,TACCO,GELARE,GAT
    TINO,GRAZIOSO,ERMETISMO,IMPORT
    ANTE,0
```

CONFUCIO

Vi proponiamo una nuova versione grafica del celebre gioco delle torri di hanoi.

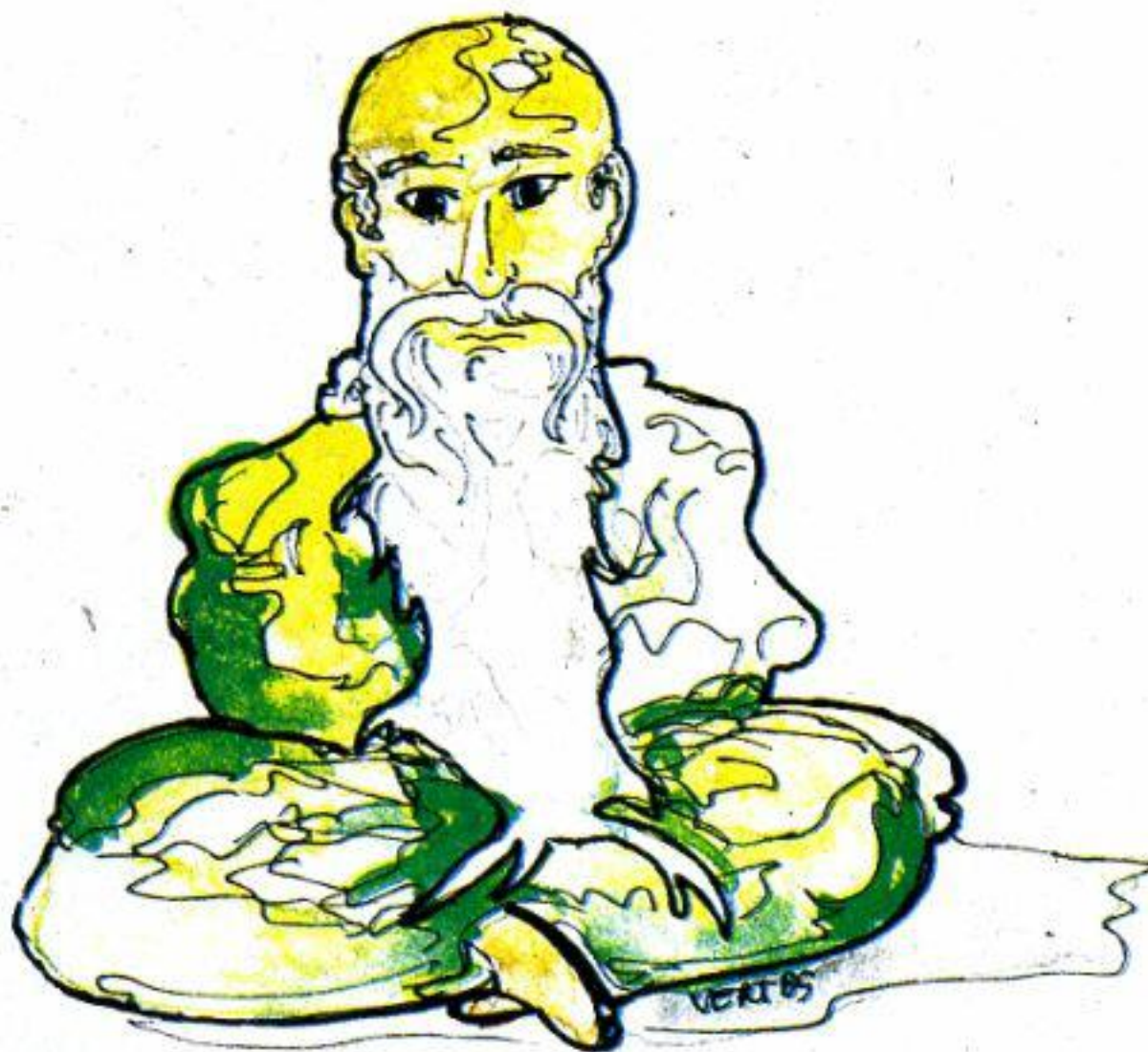
Questo game è destinato agli utenti di Commodore 16 e 64.

L'obiettivo è riuscire a trasferire tutti disconi dalla postazione numero 1 a quella contrassegnata con la cifra 2 oppure 3.

Non è consentito muovere più di una sfera alla volta ed è proibito porla al di sopra di una avente dimensioni inferiori ad essa.

Queste sono le uniche limitazioni che devono essere rispettate da chi si cimenti con questo appassionante gioco.

Il numero dei disconi può variare da un minimo di 2 ad un massimo di 7. Aumentando il numero delle sfere assisteremo ad un incremento del livello di difficoltà.

Buon divertimento.

```
4 CLR :PK=12
90 PG$="HANOI":LP=1:CR$=CHR$(13)
100 T$="[RVOFF] [RVOFF]_[RVOFF]▁[RVOFF]▂[RVOFF]▄[RVS]▀[RVS]▔[RVS]‾"
105 B$="[RVS]_[RVS]▁[RVS]▂[RVS]▄[RVOFF]▀[RVOFF]▔[RVOFF]‾[RVOFF] "
106 CC$="[NERO][ROSSO][AZZUR][VIOLA][VERDE][GIALLO][ARANC][MARR][ROSA][GRIGIO1][GRIGIO2][VERDE2][CELESTE][GRIGIO3]"
110 DIM TP$(8),T$(8),BT$(8),B$(8),CO$(13)
111 FOR I=0 TO 13:CO$(I)=MID$(CC$,I+1,1):NEXT
120 FOR I=1 TO 8
130 C$=MID$(T$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
140 TP$(I)=MID$(T$,2*I-1,1)+C$
150 C$=MID$(B$,2*I,1):C$=C$+C$+C$:C$=C$+C$+C$:C$=LEFT$(C$+C$,14)
160 BT$(I)=MID$(B$,2*I-1,1)+C$
170 NEXTI
200 L$="[RVS] [RVS]▌[RVS]▌[RVOFF]▐[RVOFF] "
210 R$="[RVOFF] [RVOFF]▌[RVOFF]▌[RVS]▐[RVS] "
220 DIM LF$(5),RT$(5)
230 FOR I=1 TO 5:LF$(I)=MID$(L$,2*I-1,2):RT$(I)=MID$(R$,2*I-1,2):NEXTI
300 CR$=CHR$(13)
490 PRINT"[CLEAR]"
500 PRINT"[DOWN]QUANTI DISCONI (MASSIMO 7) ? ";:GOSUB 60000
510 IF IN$="" THEN 500
560 N=VAL(IN$):IF N>7 THEN PRINT"[NERO]PREGO: [VIOLA]NON PIU' DI 7.[NERO]":GOTO 500
570 IF N<2 THEN PRINT"[VERDE]NON E
```

```
     SSERE REDICOLO ![NERO]":GOTO 5
     00
 900 DN$="[HOME][20 DOWN]"
 910 RT$="[14 RIGHT]"
1000 FOR I=1 TO 3:FOR J=0 TO 7:P(I,
     J)=0:NEXTJ:NEXTI
1100 PRINT"[CLEAR]";DN$;"[RVS]";:P(
     1,0)=N
1110 FOR I=1 TO 3:PRINT"[ARANC]

             ";:NEXTI
1120 PRINTDN$;"[DOWN][RVS][6 RIGHT]
     #1[11 RIGHT]#2[11 RIGHT]#3[NER
     O]"
1150 PRINTDN$;LEFT$("[7 UP]",N);
1160 FOR I=1 TO N
1170 PRINTLEFT$("[RVOFF]"+RT$,7-I+1
     );LEFT$("[RVS]L"+BT$(1),2*I+1)
     ;"_|"
1180 P(1,N-I+1)=I*2
1190 NEXTI
1800 MV=0
1900 TM=TI+60
1910 IF TI<TM THEN 1910
1950 GOSUB 8000
1960 CO=CO+1:IF CO>13 THEN CO=1
1970 CO$=CO$(CO)
2000 PR$="[NERO]DA QUALE TORRE ? ":
     GOSUB 5000:IF IN$="" THEN 1950
2010 F=VAL(IN$)
2020 IF P(F,0)<1 THEN PRINT"[VERDE2
     ]TORRE VUOTA [ROSSO]!![NERO]":
     GOTO 1900
2050 PR$="[NERO]ALLA TORRE ? ":GOSU
     B 5000:IF IN$="" THEN 1950
2060 T=VAL(IN$)
2070 IF F=T THEN PRINT"MOSSA IMPOSS
     IBILE ":GOTO 1900
2080 IF P(T,0)=0 THEN 2100
2090 IF P(F,P(F,0))>P(T,P(T,0)) THE
     N PRINT"[VERDE2]NON PUOI[NERO]
     ":GOTO 1900
```

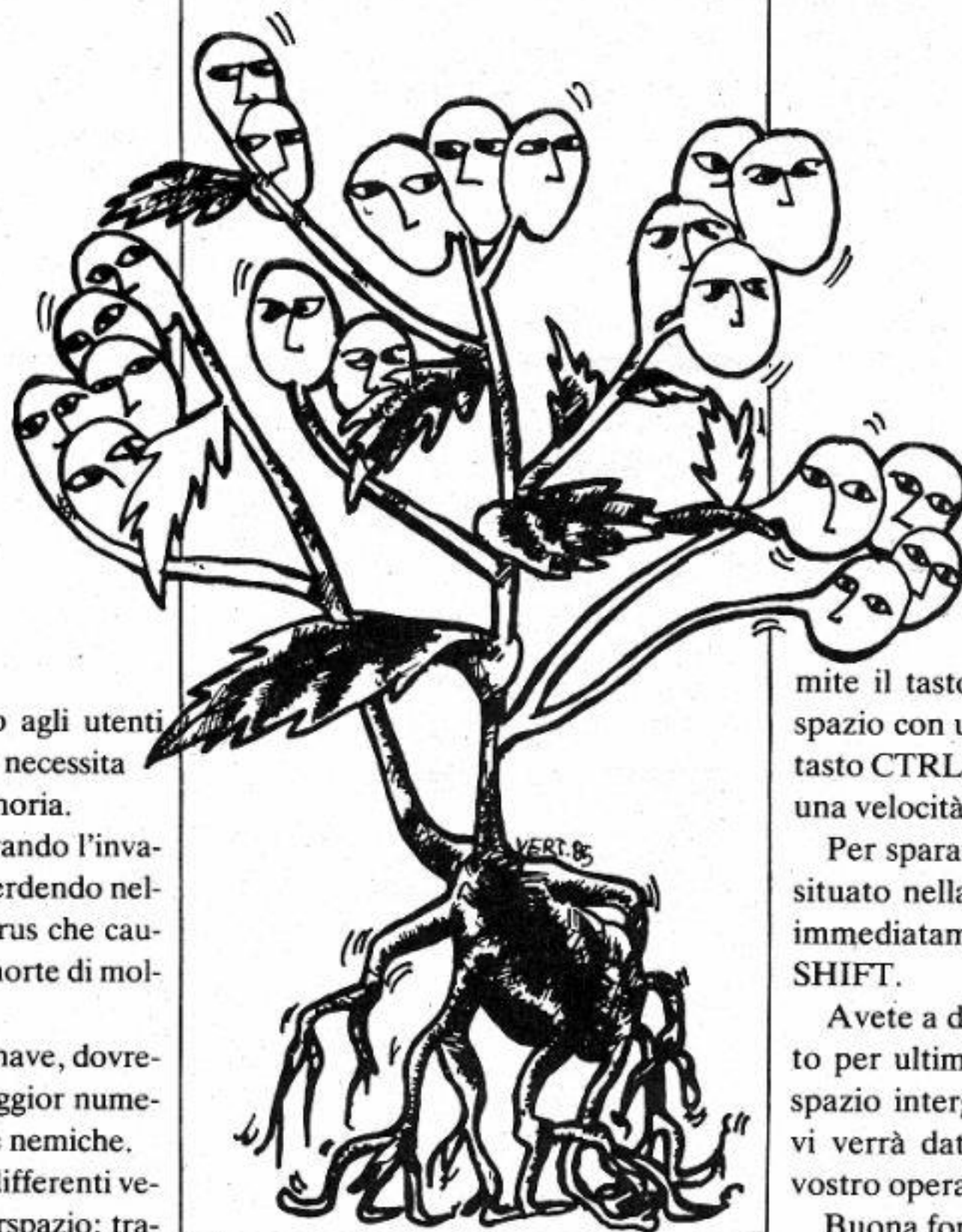
```
2100 FC=P(F,0):FW=P(F,FC):TC=P(T,0)
     :TW=P(T,TC):MV=MV+1
2110 X=1+13*(F-1)+7-FW/2
2120 PRINT DN$;LEFT$("[9 UP]",FC+1)
     ;LEFT$("[RVOFF]"+RT$+RT$+RT$,X
     );
2125 LF$=LEFT$("[DOWN][14 LEFT]",FW
     +1)
2130 FOR I=1 TO 8:T$(I)=LEFT$(TP$(I
     ),1+FW):B$(I)=LEFT$(BT$(I),1+F
     W):NEXT
2135 HT=0:J=1:IF F>T THEN J=-1
2140 FOR I=F TO T STEP J
2145 IF P(I,0)>HT THEN HT=P(I,0)
2150 NEXTI
2155 IF HT=P(T,0) THEN HT=HT+1:GOTO
      2165
2160 IF ABS(F-T)>1 THEN IF HT=P(2,0
     ) THEN HT=HT+1
2165 FOR I=FC TO HT
2170 :FOR  J=1 TO 8
2175 ::PRINT CO$;     T$(J);LF$;B$(
     J);LF$;"[2 UP]";
2180 :NEXTJ
2185 :PRINT"[UP]";
2190 NEXTI
2195 PRINT"[DOWN]";
2200 L$=LEFT$("[16 LEFT]",1+FW):R$=
     LEFT$(RT$,FW-1)
2201 PRINT"[RVS]";LEFT$("
            ",FW);"[RVOFF]";LEFT$("[
     16 LEFT]",FW);
2205 IF F>T THEN 2250
2210 FOR I=F*13 TO T*13-1
2215 :FOR J=1 TO 5
2220 PRINT CO$; LF$(J);R$;RT$(J);L$
     ;
2225 NEXTJ
2226 PRINT"[RIGHT]";
2230 NEXTI
2235 GOTO 2300
2250 FOR I=T*13 TO F*13-1
```

```
2255 PRINT"[LEFT]";
2260 :FOR J=5 TO 1 STEP -1
2265 ::PRINT CO$;    LF$(J);R$;RT$(
     J);L$;
2270 :NEXTJ
2275 NEXTI
2300 FOR I=HT-1 TO TC STEP -1
2310 :FOR J=8 TO 1 STEP -1
2320 :PRINT CO$;  T$(J);LF$;B$(J);L
     F$;"[2 UP]";
2330 :NEXTJ
2340 :PRINT"[DOWN]";
2350 NEXTI
2410 PRINT CO$;  LEFT$("[RVS]L"+BT$
     (1),FW+1);"J";  "[NERO]";
2500 P(T,0)=P(T,0)+1
2510 P(T,P(T,0))=P(F,P(F,0))
2520 P(F,0)=P(F,0)-1
2600 IF P(2,0)<>N AND P(3,0)<>N THE
     N 1900
2700 GOSUB 8000
2710 PRINT"[VERDE][3 DOWN]          
     ▄▄▄▄▄▄▄▄▄▄▄▄▄"
2720 PRINT"         [RVS]▌HAI FINITO
     [RVOFF]▌"
2730 PRINT"        "[RVS]▄▄▄▄▄▄▄▄▄▄▄
     ▄[RVOFF]""
2740 PRINT"[NERO][DOWN]HA IMPIEGATO
     [VIOLA]";MV;"[NERO]MOSSE"
2750 T=2↑N-1:PRINT"[DOWN]LA SOLUZIO
     NE PIU' VELOCE"
2760 PRINT"CORRISPONDE A ";T;"MOSSE
     ."
2770 PRINT"[DOWN]GIOCHI ANCORA? ";:
     GOSUB 60000
2780 IF LEFT$(IN$,1)<>"S" THEN END
2790  RUN
2800 END
5000 PRINTPR$;
5010 GOSUB 60000:IF IN$="" THEN RET
     URN
5015 IF LEFT$(IN$,1)="Q" THEN END
5016 IF LEN(IN$)>1 THEN 5030
5020 IF IN$>="1" AND IN$<="3" THEN
     RETURN
5030 PRINT"MA COSA DICI ????????"
5040 FOR I=1 TO 500:NEXTI
5050 PRINT"[UP]
      "
5060 PRINT"[2 UP]";PR$;LEFT$("

          ",LEN(IN$))
5070 PRINT"[UP]";:GOTO 5000
8000 PRINT"[HOME]";
8010 FOR I=1 TO 4:PRINT"[BIANCO]

          ":NEXTI
8020 PRINT"[HOME]";
8030 RETURN
9000 PRINT"[HOME]"
60000 IN$=" ":ZT=TI:ZC=2:ZD$=CHR$(20
      )
60010 GET Z$:IF  Z$<>"" THEN 60070
60020 IF ZT<=TI THEN PRINTMID$(" ▒",
      ZC,1);"[LEFT]";:ZC=3-ZC:ZT=TI+
      15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF (Z AN
      D 127)<32 THEN PRINT" [LEFT]";
      :GOTO 60110
60080 IF FL AND (Z AND 127)>64 AND (
      Z AND 127)<91 THEN Z$=CHR$((Z+
      128) AND 255)
60090 IF ZL>254 THEN 60010
60100 IN$=IN$+Z$:PRINTZ$;ZD$;Z$;
60110 IF Z=13 THEN IN$=MID$(IN$,2):P
      RINTCR$;:RETURN
60120 IF Z=20 AND ZL>1 THEN IN$=LEFT
      $(IN$,ZL-1):PRINT"[LEFT]";:GOT
      O 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>
      1)):FOR Z=2 TO ZL:PRINTZ$;:NEX
      TZ:GOTO 60000
60140 GOTO 60010
```

TRIFIDI

Nello spazio intergalattico, con il vostro Vic 20 contro gli alieni che vogliono distruggere la terra.

Questo gioco è destinato agli utenti del computer Vic 20 e non necessita di alcuna espansione di memoria.

Degli alieni stanno preparando l'invasione del pianeta terra, disperdendo nell'atmosfera un pericoloso virus che causerà, immancabilmente, la morte di molti esseri umani.

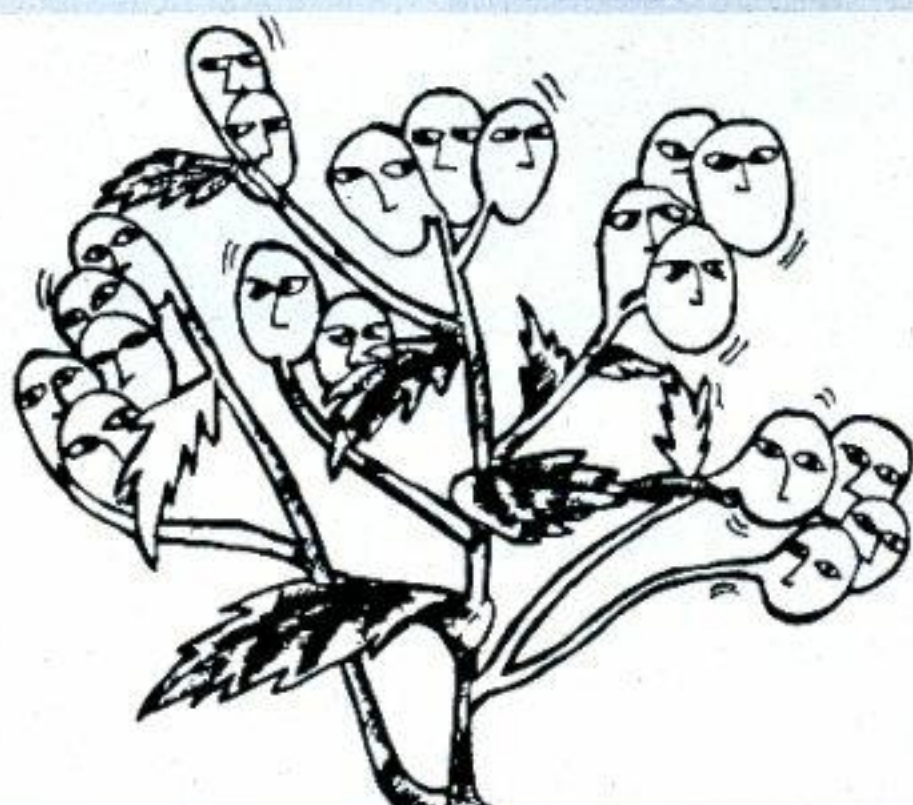
A bordo della vostra astronave, dovrete cercare di abbattere il maggior numero possibile di navi galattiche nemiche.

Avete a disposizione due differenti velocità di movimento nell'iperspazio: tramite il tasto SHIFT vi sposterete nello spazio con una velocità ridotta mentre il tasto CTRL vi permetterà di raggiungere una velocità fotonica.

Per sparare dovete avvalervi del tasto situato nella zona destra della consolle, immediatamente a sinistra del tasto SHIFT.

Avete a disposizione un tempo limitato per ultimare la vostra missione nello spazio intergalattico, terminato il quale vi verrà dato il responso definitivo sul vostro operato.

Buona fortuna.

```
1 POKE 56,24:POKE 52,24:CLR :GOS
  UB 9000:POKE 36879,8:PRINTCHR$
  (8):A=30720:D(6)=29
2 VS=36877:VI=36878:POKE VI,15:L
  =0:D(1)=-1:D(2)=-29:D(3)=-28:D
  (4)=-27:D(5)=1
3 D(7)=28:D(8)=27:W(1)=64:W(2)=7
  7:W(3)=93:W(4)=78:W(5)=64:W(6)
  =77:W(7)=93
4 POKE 36866,28:POKE 36867,72:PO
  KE 36864,7:POKE 36865,19:POKE
  648,28:PRINT"[CLEAR]"
5 POKE 648,30:PRINT"[CLEAR]":FOR
   I=0 TO 5:POKE 7674+I,32:POKE
  7674+A+I,1:NEXT:W(8)=78
6 P=7680:R=1:U1=32:U=7800:R1=1:F
  OR I=1 TO 99:POKE 7168+RND(1)*
  1008,74+RND(1)*2
7 NEXT:P1=32:FOR I=1 TO 15:POKE
  7168+RND(1)*1008,76:NEXT:TI$="
  000000"
30 T=PEEK(197):Q=0:IF T=29 THEN Q
   =-1:POKE VS-1,225
32 IF T=37 THEN Q=1:POKE VS-1,227
33 IF T=30 THEN 100
34 R=R+Q:IF R<1 THEN R=8
35 POKE VS-1,0:IF R>8 THEN R=1
36 POKE P,P1:P=P+PEEK(653)*D(R):I
   F P>8176 THEN P=P-1008
37 IF P<7168 THEN P=P+1008
38 P1=PEEK(P):POKE P,65+R:POKE P+
   A,7:IF P1=65+R1 THEN 200
40 IF RND(1)>.6 THEN R1=R1+INT(RN
   D(1)*3)-1:POKE VS,235+RND(1)*1
   8
42 IF R1<1 THEN R1=8
43 IF R1>8 THEN R1=1
```

```
44 POKE VS,0:POKE U,U1:U=U+D(R1):
   IF U>8176 THEN U=U-1008
45 IF U<7168 THEN U=U+1008
46 U1=PEEK(U):POKE U,65+R1:POKE U
   +A,5:IF U1=65+R THEN 200
49 IF TI$<"000200" THEN 30
50 PRINT"[HOME][BIANCO]TEMPO SCAD
   UTO";
51 PRINT"[2 DOWN]PUNTEGGIO:"SC:SC
   =0:FOR I=1 TO 3000:NEXT:WAIT 1
   97,63:GOTO 2
100 IF F<>0 THEN 110
102 Y=D(R):F1=R+65:F=P:L=0:K=R:G=P
    EEK(36868) OR 128
110 POKE VS-1,G:F=F+Y:F1=PEEK(F):P
    OKE F,W(K)
111 IF F<7168 OR F>8176 THEN POKE
    F,F1:POKE VS-1,0:GOTO 34
120 POKE VS-1,0:IF F1=R1+65 THEN 1
    32
121 L=L+1:IF L=12 THEN POKE F,F1:F
    =0:GOTO 34
122 POKE F,F1:GOTO 110
129 POKE U,42:FOR I=1 TO 4:FOR J=1
     TO 8:POKE U+D(J)*I,RND(1)*2+7
    4:POKE VS,220+J*3:NEXTJ,I
130 FOR I=1 TO 4:FOR J=1 TO 8:POKE
     U+D(J)*I,32:POKE VI,15-I*3:NE
    XTJ,I
131 POKE VS,0:POKE VI,15:POKE U,32
    :U=7168+INT(RND(1)*28):U1=32:F
    =0:R1=INT(RND(1)*8+1):RETURN
132 GOSUB 129:SC=SC+1:FOR I=1 TO 1
    0:POKE 7168+RND(1)*1008,RND(1)
    *2+74:NEXT:GOTO 34
200 GOSUB 129:PRINT"[HOME][BIANCO]
     COLLISIONE";:GOTO 51
9000 FOR I=0 TO 1023:POKE 6144+I,PE
     EK(32768+I):NEXT:POKE 36869,25
     4:FOR I=0 TO 87:READ A
9010 POKE 6672+I,A:NEXT:RETURN
9020 DATA 0,7,30,254,30,7,0,0,0,64,
     48,60,31,30,12,4,16,16,16,56,5
     6,124,124,68
9030 DATA 0,2,12,60,248,120,48,16,0
     ,224,120,127,120,224,0,0,16,48
     ,120,248,60,12
9040 DATA 2,0,68,124,124,56,56,16,1
     6,16,8,12,30,31,60,48,64,0,0,0
     ,8,0,0,0,0,0,0
9050 DATA 0,0,0,64,0,0,0,0,0,0,6,6,
     0,0,0
```

giochiamo con....

The day after

Questo gioco simula un'ipotetica giornata dopo lo scoppio della bomba atomica. Per salvare la terra dovete pilotare la vostra navicella per andare a recuperare delle palline rosa altamente radioattive.

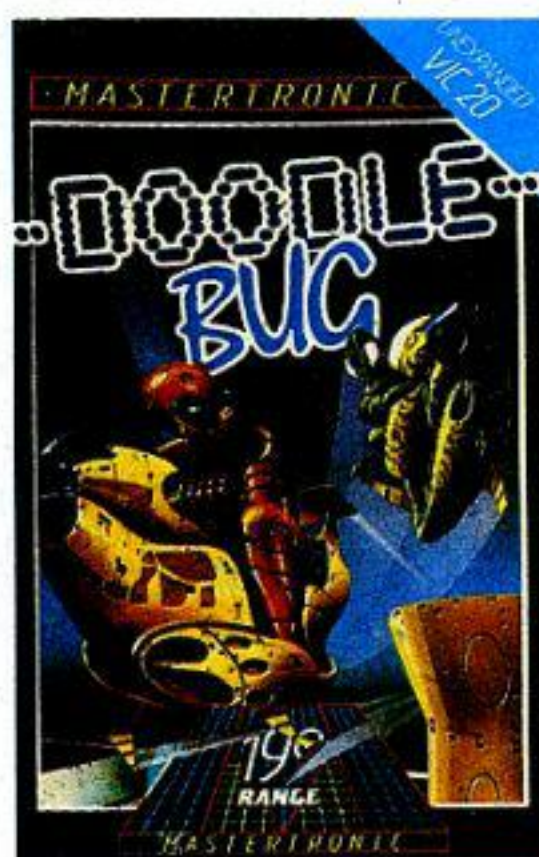
Non è un gioco facile come sembra: infatti, le persone ancora in vita (diventate pazze) cercheranno di distruggere la navicella tramite dei cannoncini. Quando sarete riusciti ad uscire dall'hangar occorre dirigersi su uno dei pianeti e ritirare le palline radioattive; fatto questo ogni volta occorre uscire dal quadro attuale e così via per 5 volte per tutti i cinque pianeti. Al termine si passerà al quadro finale: una caverna rossa dove è situata la pallina più grande di tutte e naturalmente la più pericolosa. Presa quest'ultima, il gioco ricomincia al secondo livello. Ad ogni nuovo quadro, sin da quello iniziale, c'è un'astronave che tenterà di distruggervi venendovi addosso. Per avere una spinta regolare è meglio inserire l'autofire del joystick; (se lo possedete); se invece si vuole una spinta elevata occorre disinserire l'autofire e premere manualmente il bottone del joystick.

Attenzione: dato che le immagini del gioco sono tridimensionali occorre prestare la massima attenzione durante le manovre di aggancio delle palline radioattive.

Tasti comando

Z = sinistra
X = destra
CRSR in basso = turbo reattore
CRSR destra = gancio trasporto

Idea	9
Giocabilità e suono	8
Animazione	7
Voto	8

Doodle bug

Questo gioco rappresenta una fusione tra il conosciutissimo e ormai famoso "Pac man" e la graziosa coccinella di "Lady bughi".

Si tratta sicuramente di uno dei migliori giochi per il Vic 20 della MasterTronic. Un alieno deve girare per un labirinto al fine di recuperare dei "volanti" disseminati su tutti i vari percorsi. Su alcune pareti del labirinto vi sono porte girevoli che servono sia per arrivare in un punto del labirinto senza fare un giro enorme sia per far cambiare strada all'alieno nemico.

Vi sono infatti anche degli alieni, che aumentano di quadro in quadro fino a diventare sei, e che tentano di eliminare il giocatore, finchè questo non elimina loro.

Il tuo omino generalmente è colore blu e all'inizio della partita si trova in mezzo al quadro. Tenendo premuto il bottone fire del joystick, l'omino sgancia una mina per distruggere il nemico che, però, ricomparirà subito in un punto casuale del quadro.

Voi avrete a disposizione 5 alieni e 5 aerei.

I più abili possono scegliere il numero degli alieni nemici, (mass. 5) e i quadri di gioco (mass. 6).

Idea	8
Giocabilità e suono	9
Animazione	8
Voto	8

Cattura l'

Nei giochi di adventure le capacità mentali contano di più dell'abilità a manovrare il joystick; metti alla prova la tua fantasia e la tua perseveranza, uniche qualità che ti permetteranno di arrivare alla fine del gioco.

Negli ultimi anni il mercato ha offerto un nuovo gioco (si fa per dire) al pubblico: l'adventure. Il gioco non è di concezione nuova; la novità sta nella lingua usata: finalmente compaiono gli adventures in italiano, un gioco quindi alla portata di tutti.

Per gli appassionati degli "arcades" sarà un dramma, ma di certo alla lunga fornirà un divertimanto maggiore dovuto al coinvolgimento del giocatore e non solo alla sua prontezza di riflessi.

Che cosa è un adventure? Vediamone una sintesi:

L'uomo, con in mano un idolo d'oro e delle pietre preziose, balza su una piroga nel tentativo di sfuggire agli indigeni che lo inseguno. La corrente lo trasporta verso valle, gli inseguitori però, si avvicinano sempre più. Subito si rende conto di non avere alcun mezzo di propulsione; prova con le mani: una frotta di pesci piranha gli si avvicina; disperato si gira e spara sugli inseguitori, ma questi, in gran numero, gli sono ormai addosso e lo spingono nel fiume...

Immediatamente sul video appare una didascalia...: "Sei morto, Vuoi giocare ancora (S/N)?".

Il segnale del computer è come un risveglio, l'àvventura non è finita, l'uomo nella sua fuga precipitosa si è distratto: dimenticando le pagaie ha messo la parola fine alla vicenda. Quella che doveva concludersi con allori e ricchezze si è trasformata in morte prematura, che costringe a ricominciare (dando libero sfogo alla fantasia) la ricerca di un oggetto, di un simbolo per cui valga la pena di tentare.

Si riprende il gioco, questa volta senza dimenticare niente, prestando la massima attenzione a tutto quello che si incontra sulla strada, fossero anche semplici parole.

Questo gioco a differenza dei classici "video-games", non richiede riflessi pronti e colpo d'occhio, ma notevole concentrazione fino ad immedesimarsi completamente nel personaggio; importante è capire come il suo ideatore ha sviluppato la storia, entrare nella mente, nel modo di pensare dell'autore per capire quello che gli frullava nella testa nel momento in cui scriveva la trama.

L'essenza del gioco sta nell'intricato sistema di trappole, oggetti (nascosti e

avventura!

non), enigmi e quesiti da risolvere, percorrendo strade con personaggi amici e nemici, raccogliendo oggetti che serviranno a superare un ostacolo tra i monti o la foresta o semplicemente a recuperare la chiave che apre la porta di casa.

Disegni, suoni e rumori, possono fare da cornice alla vicenda, lasciando però il più possibile libera la fantasia del giocatore nel crearsi le scene dove hanno luogo le varie azioni, il meglio è rappresentato dal niente: niente grafica, musica o rumori (opinione dello scrivente).

Un consiglio utile per chi volesse cimentarsi in un adventure è di armarsi di penna e carta, prendendo nota di tutto quello che ci sembra (ed anche quello che non ci sembra) importante. Ogni spostamento, ogni ostacolo, ogni frase (magari sibillina) è meglio annotarla, potrà tornarci utile in un altro posto e fornirci la soluzione od una traccia che ci metta sulla strada giusta. In definitiva è necessario trasformarsi in detective e un buon investigatore non può fidarsi solo della memoria, specialmente se il viaggio è lungo, complesso e ha molti dati da ricordare.

La tecnica delle annotazioni è soggettiva. Un'altro sistema potrebbe essere quello di segnare una casella per ogni spostamento nella direzione in cui è avvenuto, raccordando con una freccia e marcando ogni casella con un numero sotto cui scrivere tutto quello che ci interessa: può servirci a ricordare che in quel posto è meglio non andare o che si è lasciato un oggetto che potrà servirci in seguito.

I comandi vengono dati al computer con una o due parole, in certi casi con la sola iniziale. Possono essere dati in prima persona o rivolti ad un compagno di viaggio (a mio avviso la seconda scelta offre maggiori probabilità al giocatore di immedesimarsi nella parte).

Normalmente i comandi dati con la sola iniziale sono quelli di spostamento (Sud, Est, Nord e Ovest), Inventario (elenca il materiale recuperato durante il gioco), Aiuto (può essere fornito in aiuto per la sessione in corso), Fine (viene usata per indicare la fine del gioco ed offre la possibilità di registrare i dati e la posizione raggiunta; al termine della registrazione si può riprendere il gioco).

Altri ordini vengono dati con due parole (salvo richieste specifiche); la prima è un verbo (prendi, vuota, colpisci ecc...), la seconda indica un oggetto (pistola, corda, coltello ecc...). A questo punto l'elaboratore passa al setaccio tutte le possibilità offerte dal programma, dando una risposta generica tipo "non capisco", o una affermativa di esecuzione del comando.

Avremo quindi degli ordini tipo "prendi pistola", "getta diamanti", "togli tuta" (spaziale), etc.

Ora bando alle chiacchere, cominciamo a giocare; le difficoltà sono molte ma anche le soddisfazioni saranno pari, quindi auguri e buon divertimento.

Nel numero 7 della rivista su cassetta Commodore Club, comparirà l'adventures SOLARA che riunisce al suo interno tutti i temi trattati in questo articolo.

La potrete reperire in edicola contemporaneamente a questo numero di Commodore Computer Club.

Paola Magrin

Non lasciarti sfuggire il prossimo

Perchè anche sotto l'ombrellone potrai rinfrescarti le idee sui Data Base.

Troverai:

- **La Teoria**
- **Banca Dati. Anteprima da Commodore Club, il nostro data-base per te**
- **The Manager e SuperBase**
- **Tutto archivio, un programma tutto Basic**

K 16

IL PORTA COMPUTER CHE FA SPAZIO

Mobile di med. dens., laccato bianco, che ti permette di raggruppare tutti i componenti del tuo computer in uno spazio di cm. 64x45, eliminando al tempo stesso tutti i fili esterni.

I lettori che invieranno il coupon qui a destra, compilato in ogni sua parte, riceveranno il K16 a sole L. 85.000; oppure il set completo (esclusa la sedia) a L. 175.000; oppure solo la scrivania a L. 95.000. Spedizione postale e IVA comprese nel prezzo.

linea legno brevettato

di Amorusi G. Battista

Via Montalbano, 25L - Loc. Mercatale
Tel. (0571) 501563 - 50059 VINCI (Fi) - Italy

Il Signor ....
Via .... Città ....
desidera ricevere N. ....
☐ K 16 ☐ Set completo ☐ Scrivania
Al ricevimento della merce pagherò in contrassegno.
Firma ....

SCAMBIATEVI LE LISTE *o promuovete un Club*

Giovanni Breda - Via G. Mameli 19 - 37100 Verona - Tel. 045/46491

Raffaella Libbi - Via Firenze 17 - 64021 Giulianova - Tel. 085/8001548

Massimo Proia - Via Pubblico Passeggio 16 - 29100 Piacenza - Tel. 0523/32417

Giuseppe Moreschi - Via Paterno' 1 - 20143 Milano - Tel. 02/8134094

Nicola Mastroiacovo - Via 19 De Denominare 54 - 64029 Silvi MArina (Te) -Tel. 085/932437

Francesco Pucci - Via Perris 14 - 84012 Angri (Sa) - Tel. 081/949312

Cosimo Tantillo - Via Rizzo 15 - 90010 Aspra (Pa) - Tel. 091/930314

Alberto Frabetti - Via Ricciarelli 139 - 44100 Ferrara - Tel. 0532/61034

Stefano Barcaroli - Via Buozzi 8 - 05021 Acquasparta (Tr) - Tel. 0744/930387

Dario Pausin - Via Pascoli 24 - 34129 Trieste - Tel. 040/773766

Bruno Gandolfi - Via P. Calamandrei 1 - 14049 Nizza Monf. (At) - 0141/727216

Fabio Zerbini - Via Sacconi 5 - 20139 Milano - Tel. 02/538757

Stefano Ferreri - C.so Grosseto 259 - 10147 Torino - Tel. 011/296882

Franco Rigamonti - Casella Postale 26 - 22061 Barzago (Co)

Giuseppe Borracci - Via Mameli 15 - 33100 Udine - Tel. 0432/291665

Francesco Iozia - Via dello Stadio 4 - 97014 Ispica (Rg)

Ignazio Mannone - C.da Dammusello 504 - 91025 Marsala (Tp) - Tel. 0923/967922

Antonio Tassone - Via Montecuccoli 17 - 00176 Roma - Tel. 06/7560394

Salvatore Sciglio - Via Pirandello 33 - 98039 Taormina (Me) - Tel. 0942/23797

Dino Marocchi - Via Marconi 302 - 65100 Pescara - Tel. 085/68352

Danilo Curiazi - Via del Lavoro 25 - 40127 Bologna - Tel. 051Œ5765

Silvano Bompieri - Via Baccaglioni 8 - 46040 Monzambano (Mn) - Tel. 0376/845372

Raphael Eredità - Fermo Posta Milanofiori - 20089 Milano - Tel. 02/8257958

Gianni Mazzesi - Via Cella 329 - 48020 S. Stefano (Ra) - Tel. 0544/573529

Enrico Antinozzi - C.so Europa 26 - 80127 Napoli - Tel. 081/643119

Pietro Masi - Scalo S. Lorenzo 47 - 00185 Roma - Tel. 06/4955167

Nicola Gianno - Via Marsala 351 - 91020 Rilievo - Tel. 0923/864559

Vincenzo D'Alessandro - Via Sperone 7 - 90123 Palermo - Tel. 091/475988

Maurizio Guidato - Via Putignano 26 - 56014 Pisa - Tel. 050/982281

Alessandro Armemio - Via Cardinale Mimmi 24 - 70124 Bari - Tel. 510980

Marco Schito - Via Lucca 36 - 20152 Milano - Tel. 02/4591526

Club 64 Montagnana Software - Casella Postale 6 - 35044 Montagnana - Tel. 0429/82469

Nicola Esposito - Via MArchesa 43 - 80041 Boscoreale (Na) - Tel. 081/8591253

Gian Maria Agretti - Via Del Genio 4/5 - 40135 Bologna - Tel. 051/414630

Lorenzo Daniele - Viale Faenza 26/4 - 20142 Milano - Tel. 02/816426

Nicola De Francesco - Vicolo Sguizzante 11 - 25121 Brescia - Tel. 030/55480

Maurzio Daloso - Via Randaccio 1 - 36100 Vicenza - Tel. 0444/56236

Albano Filiaci - Via B. Miriam 61 - 63035 Offida (ap) - Tel. 0736/80144

Dario Gamba - Via Guido Reni 96/12 - 10136 Torino - Tel. 011/3090175

Emanuele Ranieri - Via S. Anna 3 - 05019 Orvieto - Tel. 0763/90475

Marco Pendino - Via Ugo Betti 25 - 20100 Milano - Tel. 02/3087174

Marzio Bruni - Via Ricrodi 11 - 20131 Milano Tel. 02/203973

Achille Lamma - Via Opicina 3 - 48100 Ravenna - Tel. 0544/420782

Francesco D'Andria - Via G. Muna 33 - 74100 Taranto - Tel. 099/314517

Claudio Giovanelli - Via Ripamonti - 20141 Milano - Tel. 02/536926

Omar Ricci - Via F. Solmano 33 - 16030 Sori (Ge) - Tel. 0183/701116

Gennaro Sessa - Via Roma 352 - 84092 Bellizzi (Sa)

Giuseppe Romano - Via Nazionale 12 - 98065 Montalbano Elicona (Me) - Tel. 0941/679040

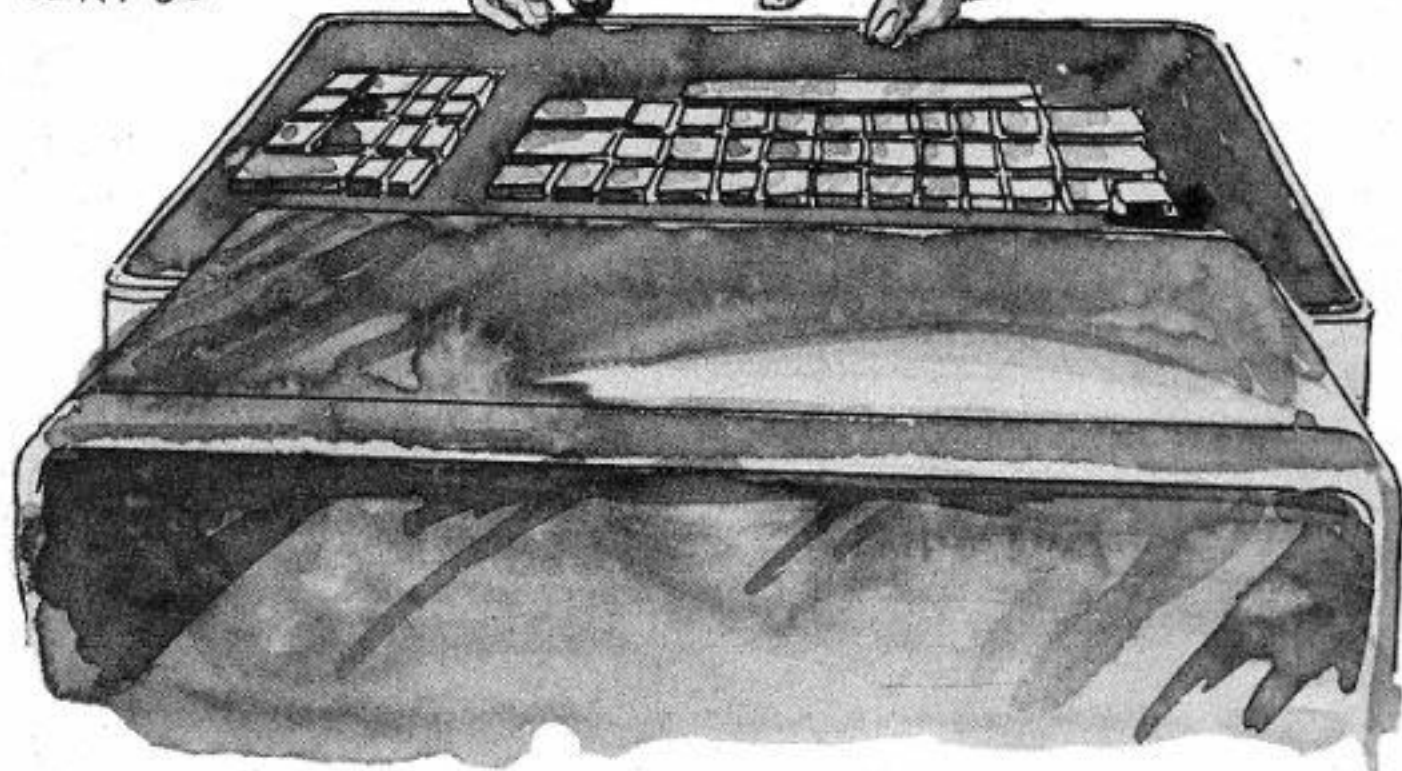
Massimo Bovi - Via G. Galilei 5 - 33170 Pordenone - Tel. 0434/42586

Roberto Manfroni - Via B. Buozzi 4 - 00049 Velletri - Tel. 06/9636898

Francesco & Paolo - P.O. Box 85 - 25010 Rivoltella D/G (Bs) - Tel. 030/9142101/ 9141303

Ugo Marozzo - Via Festo Avieno 218 - 00136 Roma - Tel. 06/3491667

Lorenzo Giavenni - Via lattuada 26 - 20135 Milano - Tel. 02/5460905

I listati della Systems Editoriale

Un'elevata percentuale dei nostri lettori è alle prime armi nel mondo dell'informatica e incontra difficoltà nel digitare i programmi da noi pubblicati.

I caratteri "speciali" bianchi su fondo nero (semi-grafici in "reverse") che rappresentano precisi comandi per i computer Commodore sono riportati nel listato di esempio a sinistra così come appaiono digitandoli su video o su stampante, mentre a destra come li rappresentiamo nei nostri listati.

La riga 360, ad esempio, deve così esser interpretata:
dopo aver battuto il carattere di virgolette (") che si ottiene premendo il tasto SHIFT insieme con il tasto 2, è necessario battere il carattere CRSR DOWN (il tasto, cioè, che normalmente sposterebbe il cursore nella cella video sottostante).

Analogamente, nella riga 180 del listato "tradotto" (di destra), il termine [NERO] sta a significare che bisogna utilizzare il carattere speciale del colore nero (tasto CTRL insieme con il tasto 1, vedi listato di sinistra).

Per ricordare in che modo vengono normalmente visualizzati i caratteri speciali,nella seconda parte delle righe di sinistra (dopo i REM) sono riportati i tasti che è necessario premere per ottenere il carattere-comando "speciale".

La Redazione

```
100 REM I CARATTERI SPECIALI
110 REM DEI COMPUTER COMMODORE
120 REM COME APPAIONO NORMALMENTE
130 REM SU VIDEO O SU CARTA.
140 REM (CTRL = TASTO CONTROL)
150 REM (CMDR = TASTO COMMODORE)
160 REM (CRSR = TASTI CURSORE)
170 :
180 PRINT"■":REM CTRL+1 NERO
190 PRINT"■":REM   "  +2 BIANCO
200 PRINT"■":REM   "  +3 ROSSO
210 PRINT"◣":REM   "  +4 AZZURRO
220 PRINT"■":REM   "  +5 PORPORA
230 PRINT"■":REM   "  +6 VERDE
240 PRINT"■":REM   "  +7 BLU
250 PRINT"■":REM   "  +8 GIALLO
260 PRINT"■":REM   "  +9 REVERSE ON
270 PRINT"■":REM   "  +0 REVERSE OFF
280 PRINT"■":REM CMDR+1 ARANCIO
290 PRINT"■":REM   "  +2 MARRONE
300 PRINT"■":REM   "  +3 ROSSO CHIARO
310 PRINT"■":REM   "  +4 GRIGIO 1
320 PRINT"■":REM   "  +5 GRIGIO 2
330 PRINT"■":REM   "  +6 VERDE CHIARO
340 PRINT"■":REM   "  +7 BLU CHIARO
350 PRINT"■":REM   "  +8 GRIGIO 3
360 PRINT"■":REM  CRSR IN BASSO
370 PRINT"■":REM  CRSR A DESTRA
380 PRINT"□":REM  CRSR IN ALTO
390 PRINT"■":REM  CRSR SINISTRA
400 PRINT"■":REM  HOME
410 PRINT"■":REM  CANCELLA SCHERMO
420 :
430 REM  ESEMPI DI VISUALIZZAZIONE:
440 PRINT"■■■■■■■":REM CANCELLA SCHERMO,
450 :                REM CRSR DWN DUE VOLTE
460 :                REM CRSR DESTRA TRE "
470 :
480 PRINT"■■■■":REM BIANCO,CRSR SINISTRA
490 :           :REM DUE VOLTE E CRSR DWN
500 :           :REM UNA SOLA VOLTA
```

```
100 REM  I CARATTERI
110 REM  SPECIALI: COME
120 REM  VENGONO INDICATI
130 REM  SULLE RIVISTE:
140 REM    COMMODORE
150 REM  E COMMODORE
160 REM  COMPUTER CLUB.
170 :
180 PRINT"[NERO]"
190 PRINT"[BIANCO]"
200 PRINT"[ROSSO]"
210 PRINT"[AZZUR]"
220 PRINT"[VIOLA]"
230 PRINT"[VERDE]"
240 PRINT"[BLEU]"
250 PRINT"[GIALLO]"
260 PRINT"[RVS]"
270 PRINT"[RVOFF]"
280 PRINT"[ARANC]"
290 PRINT"[MARR]"
300 PRINT"[ROSA]"
310 PRINT"[GRIGIO1]"
320 PRINT"[GRIGIO2]"
330 PRINT"[VERDE2]"
340 PRINT"[CELESTE]"
350 PRINT"[GRIGIO3]"
360 PRINT"[DOWN]"
370 PRINT"[RIGHT]"
380 PRINT"[UP]"
390 PRINT"[LEFT]"
400 PRINT"[HOME]"
410 PRINT"[CLEAR]"
420 :
430 REM   ESEMPI
440 PRINT"[CLEAR][2 DOWN]
    [4 RIGHT]"
450 :
460 :
470 :
480 PRINT"[BIANCO][2 LEFT
    ][DOWN]"
```

RISPOSTE

al quiz pubblicato alle pagine 32-33-34

1 - Per il computer, VIVA LA PAPPA rappresenta il nome di una variabile numerica che può essere abbreviata con VI. La risposta è pertanto: 0 (supponendo che il computer sia stato appena acceso o resettato).

2 - La risposta esatta è OUT OF MEMORY ERROR. Una subroutine non può "chiamare" se stessa. Se si tenta di fare ciò è come se si tentasse di superare il massimo numero di "nidificazioni" (=chiamata di una subroutine all'interno di un'altra subroutine) di subroutine possibile che è 38.

3 - Il cursore viene posizionato nella prima cella video in alto a sinistra. Il comando, infatti, è in fin dei conti un banale: PRINT CHR$(19).

4 - Il programma continua eseguendo, se c'è, l'istruzione successiva al comando END. In caso contrario compare nuovamente READY.

5 - Risposta C

6 - Risposta C

7 - LET è l'istruzione che assegna un valore ad una variabile e che fa parte del linguaggio arcaico Basic. Nonostante sia obsoleta, è ancora presente in tutti i linguaggi Basic. E' quindi un'istruzione superflua dato che è possibile ottenere, ad esempio, lo stesso effetto sia con:
LET A=989 sia con: A=989.

8 - Sono egualmente valide le risposte "C" e "D"

9 - La risposta esatta è la "C". A livello elettronico, infatti, il tentativo di scrivere un dato in una memoria ROM è semplicemente privo di risultati e non danneggia in alcun modo i circuiti integrati interessati all'operazione.

10 - Per Operazione si intende la singola operazione elettronico - logica (=ciclo macchina) che viene eseguita dal microprocessore ad ogni impulso proveniente dall'oscillatore al quarzo.

Un'istruzione in linguaggio macchina può richiedere anche 5 cicli macchina ed un'istruzione Basic, a sua volta, può richiedere l'esecuzione di centinaia di istruzioni in L.M.

Ecco perchè, nonostante l'apparente alta velocità (misurata in milioni di operazioni al secondo) alcune operazioni effettuate con programmi Basic risultano piuttosto lente.

Commodore 64

1 - Risposta "C".

2 - Risposta "C". E' possibile allocare sprite oltre l'indirizzo 16348 ma è necessario spostare la memoria video ricorrendo all'alterazione di alcuni puntatori. Tale tecnica, come si può immaginare, è utilizzata solo da esperti programmatori.

3 - SYS 64738

4 - L'Executive non prevede l'uso del registratore a cassette e non è possibile, quindi, caricare programmi da nastro.

5 - Esiste effettivamente un programma su nastro che "trasforma" il Commodore 64 in Sinclair Spectrum senza alcuna modifica elettrica. E' possibile utilizzarlo sia per programmi Basic che per programmi L.M. protetti oppure no.

L'unico inconveniente che si verifica è costituito dalla maggior lentezza di esecuzione dei programmi stessi dato che è necessario interpretare il codice Z-80 ed adattarlo al codice 6510.

Non esiste invece, almeno per quanto ci risulta, un programma che trasformi lo Spectrum in un Commodore 64.

Commodore 16

1 - Da 3072

2 - Risposta "A"

3 - Drive per minifloppy.

4 - GRAPHIC

5 - Rintracciare gli errori di programmazione. Può comunque essere definito a piacere dall'utente ricorrendo al comado KEY.

Vic 20

1 - Non è possibile generare sprite col Vic 20.

2 - Risposta "B"

3 - SYS 64802. Con Run/stop e Restore si ottiene il ripristino di alcuni puntatori, dell'Interrupt, del suono, del colore ecc. ma non dell'intero sistema.

4 - Risposta "D". Le prime tre operazioni sono invece perfettamente lecite.

5 - 199000 italianissime lire.

Alessandro de Simone

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 28.000 a mezzo c/c postale n° 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n ....
per l'importo di L. 28.000 intestato a Systems Editoriale

Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L............... Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐

Posseggo un C64 .... si ☐ ... no ☐
Posseggo un VIC 20 .... si ☐ ... no ☐
Posseggo un Commodore Plus 14 .... si ☐ ... no ☐
Posseggo un Commodore Plus 16 .... si ☐ ... no ☐
Posseggo un registratore dedicato .... si ☐ ... no ☐
Posseggo un drive 1541 .... si ☐ ... no ☐
Posseggo una stampante .... si ☐ ... no ☐
Posseggo un monitor .... si ☐ ... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco....................................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI ....

....

....

....

....

....

CERCO/OFFRO CONSULENZA ....

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

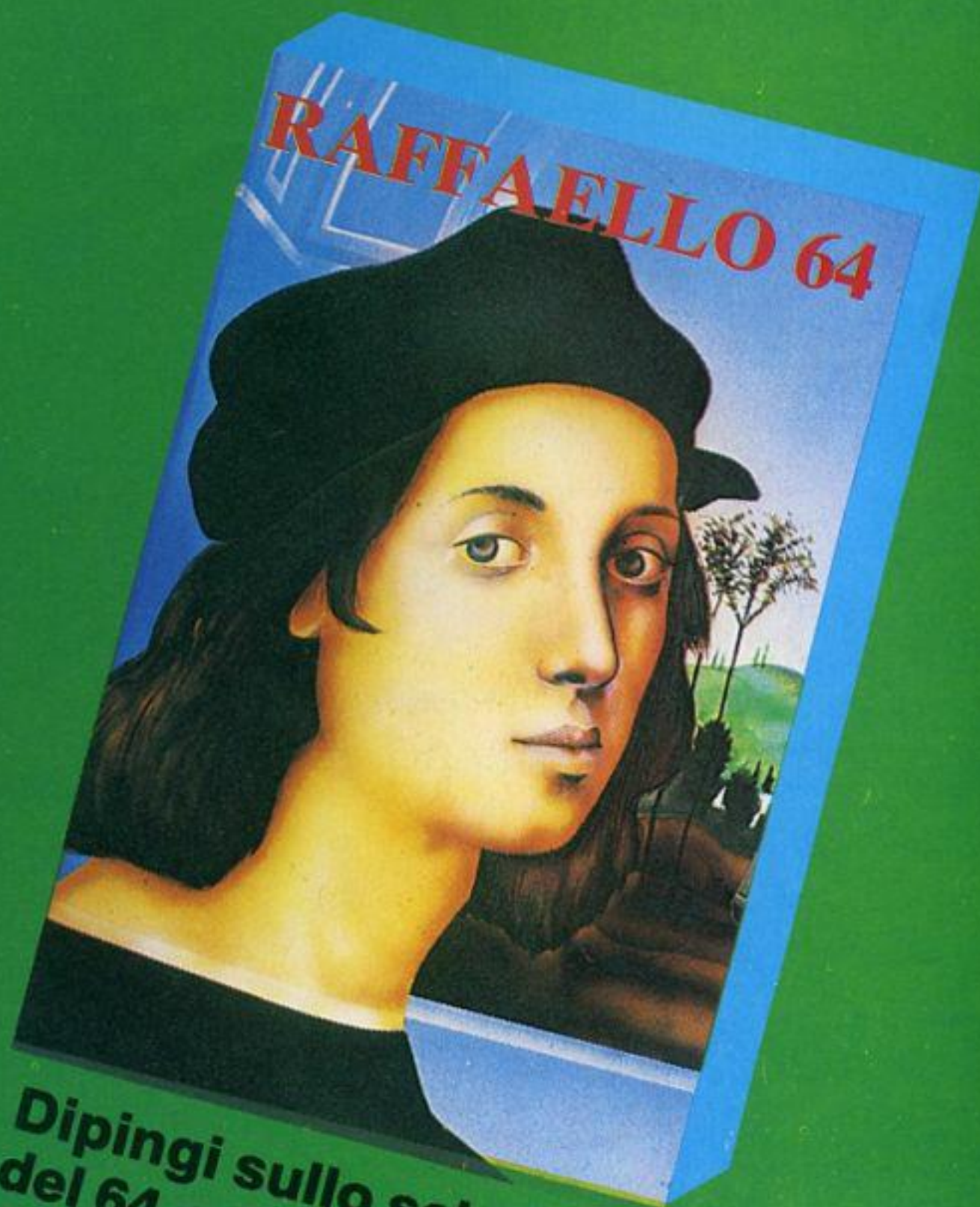
INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Due grandi software per il tuo 64

Impara il linguaggio macchina del 64 senza libri nè dispense

Dipingi sullo schermo del 64 senza tavoletta nè Koala, solo col tuo joystick.

In edicola

VIDEOREGISTRI?

VR insegna, aggiorna
ti fa toccare con mano
tutte le novità

OGNI MESE IN EDICOLA.